EAST SEARCH 09/24/2007 TLM

	Type	L #	Hits	Search Text	DBs
1	BRS	L1	1548	712/209,210,233,300.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
2	BRS	L2	238	711/123.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
3	BRS	L3	183	711/210.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB

	Type	L #	Hits	Search Text	DBs
4	BRS	L4	138	714/53.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
5	BRS	L5	803	714/710,711.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
6	BRS	L6	267	712/213.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB

	Type	L #	Hits	Search Text	DBs
7	BRS	L7	689	instruction adj exten$4	US-PGPUB; USPAT; USOCR; IBM_TDB
8	BRS	L8	474	L7 (partition$4 or split$4 or group$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
9	BRS	L9	112	L8 contiguous$4	US-PGPUB; USPAT; USOCR; IBM_TDB
10	BRS	L10	299	instruction adj extension$2	US-PGPUB; USPAT; USOCR; IBM_TDB
11	BRS	L11	223	L10 (partition$4 or split$4 or group$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
12	BRS	L12	51	L11 contiguous$4	US-PGPUB; USPAT; USOCR; IBM_TDB
13	BRS	L14	102	L10 partition$4	US-PGPUB; USPAT; USOCR; IBM_TDB
14	BRS	L17	51	L11 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
15	BRS	L19	0	L10 partitiion$4	US-PGPUB; USPAT; USOCR; IBM_TDB
16	BRS	L20	0	instruction adj extenstion$2	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
17	BRS	L25	182	711/210.ccls.	US-PGPUB; USPAT; USOCR; IBM_TDB
18	BRS	L26	77	L25 exten$4	US-PGPUB; USPAT; USOCR; IBM_TDB
19	BRS	L28	23632	((constant$2 or immediate$2) near4 separate$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
20	BRS	L29	269	L28 instruction contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
21	BRS	L30	171	L29 pointer$2	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
22	BRS	L31	170	L30 (memory or register$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
23	BRS	L32	689	instruction adj exten$4	US-PGPUB; USPAT; USOCR; IBM_TDB
24	BRS	L33	474	L32 (partition$4 or split$4 or group$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
25	BRS	L34	112	L33 contiguous$4	US-PGPUB; USPAT; USOCR; IBM_TDB
26	BRS	L36	182	711/210.ccls.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
27	BRS	L37	77	L36 exten$4	US-PGPUB; USPAT; USOCR; IBM_TDB
28	BRS	L39	1	(((constant$2 or immediate$2) near4 separate$3) near3 pointer$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
29	BRS	L40	1	(((constant$2 or immediate$2) near4 separate$3) near4 pointer$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
30	BRS	L41	170	L31 stor$4	US-PGPUB; USPAT; USOCR; IBM_TDB
31	BRS	L42	299	instruction adj extension$2	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
32	BRS	L43	223	L42 (partition$4 or split$4 or group$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
33	BRS	L44	51	L43 contiguous$4	US-PGPUB; USPAT; USOCR; IBM_TDB
34	BRS	L45	102	L42 partition$4	US-PGPUB; USPAT; USOCR; IBM_TDB
35	BRS	L48	0	L42 partitiion$4	US-PGPUB; USPAT; USOCR; IBM_TDB
36	BRS	L49	803	714/710.ccls. or 714/711.ccls.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
37	BRS	L50	92	L49 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
38	BRS	L51	90	L50 memory	US-PGPUB; USPAT; USOCR; IBM_TDB
39	BRS	L52	177	(memory adj patch$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
40	BRS	L55	51	L43 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
41	BRS	L59	54	"L323" contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
42	BRS	L61	1018	microsequencer	US-PGPUB; USPAT; USOCR; IBM_TDB
43	BRS	L62	218	L61 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
44	BRS	L63	187	L62 memory	US-PGPUB; USPAT; USOCR; IBM_TDB
45	BRS	L64	39	L52 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
46	BRS	L65	0	instruction adj extenstion$2	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
47	BRS	L66	54	instruction near2 cache near2 width	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
48	BRS	L67	59	instruction near2 cache near2 width	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
49	BRS	L68	27	(instruction near2 cache near2 width) and instruction near2 (extension or prefix or predicate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB

	Type	L #	Hits	Search Text	DBs
50	BRS	L69	153	((combin$3 or merg$3) with instruction with (extension))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
51	BRS	L73	104	(instruction near2 cache) and ((combin$3 or merg$3) with instruction with (extension))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
52	BRS	L18	1	L15 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
53	BRS	L56	1	L46 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
54	BRS	L60	1	"6308258".pn.	US-PGPUB; USPAT; USOCR; IBM_TDB
55	BRS	L72	1	(instruction near2 cache near2 width) same instruction near2 (extension or prefix or predicate)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
56	BRS	L13	85	L9 not L12	US-PGPUB; USPAT; USOCR; IBM_TDB
57	BRS	L15	42	L14 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
58	BRS	L16	9	L12 not L15	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
59	BRS	L21	6	gschwind-michael.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
60	BRS	L22	4	altman-erik.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
61	BRS	L23	47	luick-david-a.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
62	BRS	L24	14	L23 exten$4	US-PGPUB; USPAT; USOCR; IBM_TDB
63	BRS	L27	11	L26 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
64	BRS	L35	4	altman-erik.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
65	BRS	L38	11	L37 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
66	BRS	L46	42	L45 contiguous	US-PGPUB; USPAT; USOCR; IBM_TDB
67	BRS	L47	9	L44 not L46	US-PGPUB; USPAT; USOCR; IBM_TDB
68	BRS	L53	47	luick-david-a.in.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
69	BRS	L54	14	L53 exten$4	US-PGPUB; USPAT; USOCR; IBM_TDB
70	BRS	L57	85	L34 not L44	US-PGPUB; USPAT; USOCR; IBM_TDB
71	BRS	L58	6	gschwind-michael.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
72	BRS	L70	34	((combin$3 or merg$3) near4 instruction near4 (extension))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB

	Type	L #	Hits	Search Text	DBs
73	BRS	L71	5	(instruction near2 cache) same ((combin$3 or merg$3) with instruction with (extension))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
74	BRS	L74	13	(instruction near2 cache near2 width) and instruction near2 (extension)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB
75	BRS	L75	65	instruction near2 extension near2 cache	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB

	Type	L #	Hits	Search Text	DBs
1	BRS	L2	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
2	BRS	L6	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
3	BRS	L7	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
4	BRS	L8	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
5	BRS	L9	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
6	BRS	L10	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
7	BRS	L11	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
8	BRS	L12	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
9	BRS	L17	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
10	BRS	L18	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
11	BRS	L19	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
12	BRS	L23	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
13	BRS	L24	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
14	BRS	L25	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
15	BRS	L26	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
16	BRS	L27	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
17	BRS	L28	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
18	BRS	L29	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
19	BRS	L30	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
20	BRS	L31	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
21	BRS	L32	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
22	BRS	L33	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
23	BRS	L47	506	partition$4 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
24	BRS	L51	339	(memory adj pag$4) (extend$2 near2 bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
25	BRS	L53	0	(memory adj pag$4) (extens$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
26	BRS	L54	23	(memory adj pag$4) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
27	BRS	L57	357	(32-bit adj instruction) (8-bit)	US-PGPUB; USPAT; USOCR; IBM_TDB
28	BRS	L69	2	(fixed adj length)(bit adj instruction) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
29	BRS	L77	0	"566510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
30	BRS	L85	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
31	BRS	L86	117	L85 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
32	BRS	L91	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
33	BRS	L97	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
34	BRS	L98	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
35	BRS	L99	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
36	BRS	L100	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
37	BRS	L103	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
38	BRS	L114	0	(memory adj pag$4) (extens$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
39	BRS	L115	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
40	BRS	L117	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
41	BRS	L118	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
42	BRS	L119	506	partition$4 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
43	BRS	L122	23	(memory adj pag$4) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
44	BRS	L123	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
45	BRS	L124	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
46	BRS	L126	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
47	BRS	L127	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
48	BRS	L128	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
49	BRS	L129	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
50	BRS	L130	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
51	BRS	L131	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
52	BRS	L132	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
53	BRS	L133	339	(memory adj pag$4) (extend$2 near2 bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
54	BRS	L134	357	(32-bit adj instruction) (8-bit)	US-PGPUB; USPAT; USOCR; IBM_TDB
55	BRS	L135	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
56	BRS	L136	2	(fixed adj length) (bit adj instruction) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
57	BRS	L137	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
58	BRS	L138	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
59	BRS	L140	0	"566510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
60	BRS	L168	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
61	BRS	L199	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
62	BRS	L200	117	L199 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
63	BRS	L208	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
64	BRS	L245	264	(page adj partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
65	BRS	L250	4	altman-erik.in.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
66	BRS	L251	6	gschwind-michael.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
67	BRS	L252	0	luick-david.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
68	BRS	L253	0	luick-david-.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
69	BRS	L254	47	luick-david-a.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
70	BRS	L255	0	prener-daniel.in.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
71	BRS	L256	39	prener.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
72	BRS	L257	0	rivers-jude.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
73	BRS	L260	2	sathaye-sumedh.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
74	BRS	L261	25	wellman-john-david.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
75	BRS	L262	1	L259 ((code adj page$2) or codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
76	BRS	L263	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
77	BRS	L264	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
78	BRS	L265	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
79	BRS	L267	0	(memory adj pag$4) (extens$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
80	BRS	L268	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
81	BRS	L269	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
82	BRS	L270	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
83	BRS	L271	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
84	BRS	L272	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
85	BRS	L273	0	"566510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
86	BRS	L274	2	(fixed adj length) (bit adj instruction) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
87	BRS	L275	357	(32-bit adj instruction) (8-bit)	US-PGPUB; USPAT; USOCR; IBM_TDB
88	BRS	L276	23	(memory adj pag$4) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
89	BRS	L277	0	(memory adj pag$4) (extens$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
90	BRS	L278	339	(memory adj pag$4) (extend$2 near2 bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
91	BRS	L279	506	partition$4 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
92	BRS	L280	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
93	BRS	L281	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
94	BRS	L282	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
95	BRS	L283	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
96	BRS	L284	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
97	BRS	L285	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
98	BRS	L286	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
99	BRS	L287	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
100	BRS	L288	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
101	BRS	L289	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
102	BRS	L290	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
103	BRS	L291	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
104	BRS	L292	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
105	BRS	L293	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
106	BRS	L294	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
107	BRS	L295	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
108	BRS	L296	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
109	BRS	L297	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
110	BRS	L298	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
111	BRS	L299	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
112	BRS	L304	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
113	BRS	L306	357	(32-bit adj instruction) (8-bit)	US-PGPUB; USPAT; USOCR; IBM_TDB
114	BRS	L307	339	(memory adj pag$4) (extend$2 near2 bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
115	BRS	L308	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
116	BRS	L309	23	(memory adj pag$4) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
117	BRS	L310	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
118	BRS	L311	1404	"L11" extension	US-PGPUB; USPAT; USOCR; IBM_TDB
119	BRS	L312	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
120	BRS	L313	506	partition$4 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
121	BRS	L314	836	(page near3 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
122	BRS	L315	0	((code adj page$2) near4 plit$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
123	BRS	L316	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
124	BRS	L317	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
125	BRS	L322	0	"566510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
126	BRS	L323	0	((codepage$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
127	BRS	L324	202	(codepage$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
128	BRS	L325	0	((codepage$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
129	BRS	L326	2	(fixed adj length)(bit adj instruction) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
130	BRS	L59	1	(32-bit adj instruction) (8-bit adj extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
131	BRS	L68	1	(fixed adj length) (bit adj instruction) (extension adj bits) (memory adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
132	BRS	L78	1	"5666510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
133	BRS	L79	1	"5935237".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
134	BRS	L80	1	"6314504".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
135	BRS	L109	1	"6314504".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
136	BRS	L116	1	(fixed adj length)(bit adj instruction) (extension adj bits) (memory adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
137	BRS	L121	1	"5935237".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
138	BRS	L125	1	"5666510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
139	BRS	L139	1	(32-bit adj instruction) (8-bit adj extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
140	BRS	L300	1	"5666510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
141	BRS	L303	1	"5935237".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
142	BRS	L318	1	"6314504".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
143	BRS	L319	1	"6314504".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
144	BRS	L320	1	(fixed adj length)(bit adj instruction) (extension adj bits) (memory adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
145	BRS	L321	1	(32-bit adj instruction) (8-bit adj extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
146	BRS	L329	1	(fixed adj length) (bit adj instruction) (extension adj bits) (memory adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
147	BRS	L330	1	"5935237".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
148	BRS	L331	1	"5666510".PN.	US-PGPUB; USPAT; USOCR; IBM_TDB
149	BRS	L332	1	(32-bit adj instruction) (8-bit adj extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
150	BRS	L1	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
151	BRS	L3	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
152	BRS	L4	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
153	BRS	L5	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
154	BRS	L13	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
155	BRS	L14	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
156	BRS	L15	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
157	BRS	L16	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
158	BRS	L21	36	L20 processor	US-PGPUB; USPAT; USOCR; IBM_TDB
159	BRS	L22	12	L21 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
160	BRS	L34	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
161	BRS	L35	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
162	BRS	L36	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
163	BRS	L37	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
164	BRS	L38	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
165	BRS	L39	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
166	BRS	L40	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
167	BRS	L41	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
168	BRS	L43	36	L42 processor	US-PGPUB; USPAT; USOCR; IBM_TDB
169	BRS	L44	12	L43 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
170	BRS	L45	2	L43 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
171	BRS	L46	6	L32 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
172	BRS	L48	3	partition$4 near5 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
173	BRS	L49	6	partition$4 near8 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
174	BRS	L50	29	(memory adj pag$4) near5 partition	US-PGPUB; USPAT; USOCR; IBM_TDB
175	BRS	L52	39	(memory adj pag$4) (extend$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
176	BRS	L55	59	(memory adj pag$4) (extens$4 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
177	BRS	L56	7	L55 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
178	BRS	L58	4	(32-bit adj instruction) (8-bit adj extension)	US-PGPUB; USPAT; USOCR; IBM_TDB
179	BRS	L60	4	(32-bit adj instruction) (8-bit near3 extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
180	BRS	L61	5	(32-bit adj instruction) (8-bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
181	BRS	L62	86	(32-bit adj instruction) (bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
182	BRS	L63	12	(32-bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
183	BRS	L64	101	(bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
184	BRS	L65	83	(bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
185	BRS	L66	44	(fixed adj length) (bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
186	BRS	L67	43	(fixed adj length)(bit adj instruction) (extension adj bits) page	US-PGPUB; USPAT; USOCR; IBM_TDB
187	BRS	L70	6	(fixed adj length adj instruction$2) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
188	BRS	L71	8	(fixed adj length adj instruction$2) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
189	BRS	L74	57	(extension adj bits) processor (fixed adj length)	US-PGPUB; USPAT; USOCR; IBM_TDB
190	BRS	L75	32	L73 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
191	BRS	L76	25	L75 RISC	US-PGPUB; USPAT; USOCR; IBM_TDB
192	BRS	L81	9	("4679140" | "4876639" | "4893235" | "5666510" | "5680567" | "5687344" | "5809274").PN. OR ("6314504").URPN.	US-PGPUB; USPAT; USOCR
193	BRS	L82	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
194	BRS	L83	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
195	BRS	L84	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
196	BRS	L87	36	L86 processor	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
197	BRS	L88	2	L87 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
198	BRS	L89	6	L85 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
199	BRS	L90	3	partition$4 near5 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
200	BRS	L92	44	(fixed adj length) (bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
201	BRS	L93	43	(fixed adj length)(bit adj instruction) (extension adj bits) page	US-PGPUB; USPAT; USOCR; IBM_TDB
202	BRS	L94	6	(fixed adj length adj instruction$2) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
203	BRS	L102	36	L101 processor	US-PGPUB; USPAT; USOCR; IBM_TDB
204	BRS	L104	12	L102 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
205	BRS	L105	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
206	BRS	L106	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
207	BRS	L107	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
208	BRS	L108	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
209	BRS	L110	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
210	BRS	L111	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
211	BRS	L112	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
212	BRS	L113	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
213	BRS	L120	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
214	BRS	L141	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
215	BRS	L142	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
216	BRS	L143	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
217	BRS	L144	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
218	BRS	L145	12	L87 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
219	BRS	L146	29	(memory adj pag$4) near5 partition	US-PGPUB; USPAT; USOCR; IBM_TDB
220	BRS	L147	6	partition$4 near8 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
221	BRS	L148	39	(memory adj pag$4) (extend$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
222	BRS	L149	59	(memory adj pag$4) (extens$4 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
223	BRS	L150	7	L149 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
224	BRS	L151	4	(32-bit adj instruction) (8-bit adj extension)	US-PGPUB; USPAT; USOCR; IBM_TDB
225	BRS	L152	5	(32-bit adj instruction) (8-bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
226	BRS	L153	4	(32-bit adj instruction) (8-bit near3 extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
227	BRS	L154	12	(32-bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
228	BRS	L155	86	(32-bit adj instruction) (bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
229	BRS	L156	101	(bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
230	BRS	L157	83	(bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
231	BRS	L158	8	(fixed adj length adj instruction$2) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
232	BRS	L159	32	L95 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
233	BRS	L160	25	L159 RISC	US-PGPUB; USPAT; USOCR; IBM_TDB
234	BRS	L161	9	("4679140" | "4876639" | "4893235" | "5666510" | "5680567" | "5687344" | "5809274").PN. OR ("6314504").URPN.	US-PGPUB; USPAT; USOCR
235	BRS	L163	57	(extension adj bits) processor (fixed adj length)	US-PGPUB; USPAT; USOCR; IBM_TDB
236	BRS	L164	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
237	BRS	L165	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
238	BRS	L166	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
239	BRS	L167	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
240	BRS	L170	36	L169 processor	US-PGPUB; USPAT; USOCR; IBM_TDB
241	BRS	L171	12	L170 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
242	BRS	L172	2	L170 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
243	BRS	L173	6	L168 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
244	BRS	L174	3	partition$4 near5 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
245	BRS	L175	6	partition$4 near8 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
246	BRS	L176	29	(memory adj pag$4) near5 partition	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
247	BRS	L177	39	(memory adj pag$4) (extend$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
248	BRS	L178	59	(memory adj pag$4) (extens$4 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
249	BRS	L179	7	L178 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
250	BRS	L180	4	(32-bit adj instruction) (8-bit adj extension)	US-PGPUB; USPAT; USOCR; IBM_TDB
251	BRS	L181	4	(32-bit adj instruction) (8-bit near3 extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
252	BRS	L182	5	(32-bit adj instruction) (8-bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
253	BRS	L183	86	(32-bit adj instruction) (bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
254	BRS	L184	12	(32-bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
255	BRS	L185	101	(bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
256	BRS	L186	83	(bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
257	BRS	L187	44	(fixed adj length) (bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
258	BRS	L188	43	(fixed adj length) (bit adj instruction) (extension adj bits) page	US-PGPUB; USPAT; USOCR; IBM_TDB
259	BRS	L189	6	(fixed adj length adj instruction$2) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
260	BRS	L190	8	(fixed adj length adj instruction$2) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
261	BRS	L191	57	(extension adj bits) processor (fixed adj length)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
262	BRS	L193	32	L192 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
263	BRS	L194	25	L193 RISC	US-PGPUB; USPAT; USOCR; IBM_TDB
264	BRS	L195	9	("4679140" | "4876639" | "4893235" | "5666510" | "5680567" | "5687344" | "5809274").PN. OR ("6314504").URPN.	US-PGPUB; USPAT; USOCR
265	BRS	L196	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
266	BRS	L197	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
267	BRS	L198	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
268	BRS	L201	36	L200 processor	US-PGPUB; USPAT; USOCR; IBM_TDB
269	BRS	L202	2	L201 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
270	BRS	L203	6	L199 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
271	BRS	L204	3	partition$4 near5 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
272	BRS	L205	44	(fixed adj length)(bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
273	BRS	L206	43	(fixed adj length)(bit adj instruction) (extension adj bits) page	US-PGPUB; USPAT; USOCR; IBM_TDB
274	BRS	L207	6	(fixed adj length adj instruction$2) (extension adj bits) (page adj table$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
275	BRS	L210	36	L209 processor	US-PGPUB; USPAT; USOCR; IBM_TDB
276	BRS	L211	12	L210 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
277	BRS	L212	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
278	BRS	L213	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
279	BRS	L214	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
280	BRS	L215	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
281	BRS	L216	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
282	BRS	L217	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
283	BRS	L218	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
284	BRS	L219	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
285	BRS	L220	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
286	BRS	L221	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
287	BRS	L222	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
288	BRS	L223	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
289	BRS	L224	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
290	BRS	L225	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
291	BRS	L226	12	L201 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
292	BRS	L227	29	(memory adj pag$4) near5 partition	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
293	BRS	L228	6	partition$4 near8 (individual adj page$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
294	BRS	L229	39	(memory adj pag$4) (extend$2 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
295	BRS	L230	59	(memory adj pag$4) (extens$4 adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
296	BRS	L231	7	L230 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
297	BRS	L232	4	(32-bit adj instruction) (8-bit adj extension)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
298	BRS	L233	4	(32-bit adj instruction) (8-bit near3 extend$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
299	BRS	L234	12	(32-bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
300	BRS	L235	5	(32-bit adj instruction) (8-bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
301	BRS	L236	86	(32-bit adj instruction) (bit near3 extens$3)	US-PGPUB; USPAT; USOCR; IBM_TDB
302	BRS	L237	101	(bit adj instruction) (extension adj bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
303	BRS	L238	83	(bit adj instruction) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
304	BRS	L239	8	(fixed adj length adj instruction$2) (extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
305	BRS	L241	32	L240 TLB	US-PGPUB; USPAT; USOCR; IBM_TDB
306	BRS	L242	25	L241 RISC	US-PGPUB; USPAT; USOCR; IBM_TDB
307	BRS	L243	9	("4679140" | "4876639" | "4893235" | "5666510" | "5680567" | "5687344" | "5809274").PN. OR ("6314504").URPN.	US-PGPUB; USPAT; USOCR
308	BRS	L244	57	(extension adj bits) processor (fixed adj length)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
309	BRS	L266	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
310	BRS	L301	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
311	BRS	L302	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
312	BRS	L305	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
313	BRS	L327	36	L246 processor	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
314	BRS	L328	12	L327 boundary	US-PGPUB; USPAT; USOCR; IBM_TDB
315	BRS	L333	3	((code adj page$2) near4 split$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
316	BRS	L334	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
317	BRS	L335	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
318	BRS	L336	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
319	BRS	L337	25	((code adj page$2) near4 section$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
320	BRS	L338	14	((code adj page$2) near4 divide$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
321	BRS	L339	6	((code adj page$2) near4 partition$4)	US-PGPUB; USPAT; USOCR; IBM_TDB
322	BRS	L20	117	L18 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
323	BRS	L42	117	L32 extension	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
324	BRS	L101	117	L100 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
325	BRS	L162	117	L85 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
326	BRS	L169	117	L168 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
327	BRS	L209	117	L208 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
328	BRS	L246	117	L245 extension	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
329	BRS	L247	117	L199 extension	US-PGPUB; USPAT; USOCR; IBM_TDB
330	BRS	L259	127	712/210.ccls or 712/209.ccls. (instruction adj word$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
331	BRS	L73	360	(extension adj bits) processor	US-PGPUB; USPAT; USOCR; IBM_TDB
332	BRS	L95	360	(extension adj bits) processor	US-PGPUB; USPAT; USOCR; IBM_TDB
333	BRS	L192	360	(extension adj bits) processor	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
334	BRS	L240	360	(extension adj bits) processor	US-PGPUB; USPAT; USOCR; IBM_TDB
335	BRS	L72	520	(extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
336	BRS	L96	520	(extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
337	BRS	L248	520	(extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB
338	BRS	L249	520	(extension adj bits)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	L #	Hits	Search Text	DBs
339	BRS	L258	649	rivers.in.	US-PGPUB; USPAT; USOCR; IBM_TDB
340	BRS	L340	1	combin$3.clm. and instruction.clm. and word.clm. and code.clm. and page.clm. and partition$3.clm.	US-PGPUB

	Type	Ref #	Hits	Search Text	DBs
1	BRS	S1	24	(instruction adj unit) (code adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
2	BRS	S2	13	S1 (page adj table)	US-PGPUB; USPAT; USOCR; IBM_TDB
3	BRS	S3	24	(instruction adj unit) (code adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
4	BRS	S4	13	S3 (page adj table)	US-PGPUB; USPAT; USOCR; IBM_TDB
5	BRS	S5	2	S4 width	US-PGPUB; USPAT; USOCR; IBM_TDB
6	BRS	S6	13	S4 length	US-PGPUB; USPAT; USOCR; IBM_TDB
7	BRS	S7	13	S4 size	US-PGPUB; USPAT; USOCR; IBM_TDB
8	BRS	S8	0	S4 n-bits	US-PGPUB; USPAT; USOCR; IBM_TDB
9	BRS	S9	1	S4 (n near3 bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
10	BRS	S10	13	S4 variable	US-PGPUB; USPAT; USOCR; IBM_TDB
11	BRS	S11	10	S4 (variable adj size)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	Ref #	Hits	Search Text	DBs
12	BRS	S12	0	S4 ((variable adj size) near instruction)	US-PGPUB; USPAT; USOCR; IBM_TDB
13	BRS	S13	0	S4 ((variable adj size) near3 instruction$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
14	BRS	S14	24	(instruction adj unit) (code adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
15	BRS	S15	24	(instruction adj unit) (code adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
16	BRS	S24	13	S18 variable	US-PGPUB; USPAT; USOCR; IBM_TDB
17	BRS	S17	24	(instruction adj unit) (code adj page)	US-PGPUB; USPAT; USOCR; IBM_TDB
18	BRS	S20	13	S18 length	US-PGPUB; USPAT; USOCR; IBM_TDB
19	BRS	S21	13	S18 size	US-PGPUB; USPAT; USOCR; IBM_TDB
20	BRS	S22	0	S18 n-bits	US-PGPUB; USPAT; USOCR; IBM_TDB
21	BRS	S26	0	S18 ((variable adj size) near instruction)	US-PGPUB; USPAT; USOCR; IBM_TDB
22	BRS	S27	0	S18 ((variable adj size) near3 instruction$2)	US-PGPUB; USPAT; USOCR; IBM_TDB

	Type	Ref #	Hits	Search Text	DBs
23	BRS	S23	1	S18 (n near3 bit$2)	US-PGPUB; USPAT; USOCR; IBM_TDB
24	BRS	S16	13	S14 (page adj table)	US-PGPUB; USPAT; USOCR; IBM_TDB
25	BRS	S18	13	S17 (page adj table)	US-PGPUB; USPAT; USOCR; IBM_TDB
26	BRS	S19	2	S18 width	US-PGPUB; USPAT; USOCR; IBM_TDB
27	BRS	S25	10	S18 (variable adj size)	US-PGPUB; USPAT; USOCR; IBM_TDB
28	BRS	S28	9	("5666510").URPN.	USPAT

EIC NPL SEARCH TLM 09/24/2007
10/720,585

```
Set     Items   Description
S1   12846120   COMMAND? ? OR INSTRUCTION? ? OR PROGRAM? OR PROGRAMME?
? OR
                 CODE? OR CODING? OR FUNCTION?
S2     725270   S1(5N)(MERG??? OR FUSE? ? OR FUSING OR UNIFY? OR
UNIFIE? ?
                OR UNITE? OR SYNTHESI? OR COMBIN? OR INTEGRAT? OR INCLU?
OR I-
                NCORPORAT?)
S3    2518375   ENTEN? OR PREDICAT? OR PREFIX? OR PRE()(FIX???) OR
SUFFIX?
                OR MMX OR MODIF?
S4      42784   S2 AND S3
S5        839   S4(5N)(STORE? ? OR STORING OR SAVE? ? OR ACCUMULAT? OR
KEE-
                P??? OR WRIT??? OR UPDAT?)
S6    1372702   BUFFER? OR MEMOR? OR CACHE? OR CACHING?? OR QUEUE?
S7      66406   S6(3N)(SPECIF? OR INDICAT? OR DESIR? OR REQUIR? OR
NECESS?
                OR CERTAIN?)
S8     167321   (WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR
BITS))(5N)(EQUAL? -
                OR AT()LEAST OR COMPARABL? OR IDENTICAL? OR EQUIVALEN? OR
SAME
                 OR SIMILAR)
S9      33797   (WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR
BITS))(5N)(MATCH? -
                OR AGREE? OR ALIKE OR AKIN OR CONGRUEN? OR COMMON? OR
INCOMMO-
                N?)
S10    124486   S1(5N)(WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR BITS))
S11     16283   S3(5N)(WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR BITS))
S12         0   S5 AND S6:S7 AND S8:S9 AND S10 AND S11
S13         0   S5 AND S6:S7 AND S8:S9 AND S1 AND S3
S14    610678   S1:S2 AND S3
S15     13547   S14(5N)(STORE? ? OR STORING OR SAVE? ? OR ACCUMULAT? OR
KE-
                EP??? OR WRIT??? OR UPDAT?)
S16        27   S15 AND S6:S7 AND S8:S9
S17         4   S16 AND S10 AND S11
S18        27   S16 AND S1 AND S3
S19        23   S18 NOT S17
S20        11   RD  (unique items)
S21         6   S6 AND S8:S9 AND S10 AND S11
S22         2   S21 NOT S16
File   2:INSPEC 1898-2007/Sep W3
         (c) 2007 Institution of Electrical Engineers
File   6:NTIS 1964-2007/Oct W1
         (c) 2007 NTIS, Intl Cpyrght All Rights Res
File   8:Ei Compendex(R) 1884-2007/Sep W3
         (c) 2007 Elsevier Eng.  Info. Inc.
File  34:SCISEARCH(R) CITED REF SCI 1990-2007/SEP W4
         (c) 2007 THE THOMSON CORP
File  35:Dissertation Abs Online 1861-2007/Jul
         (c) 2007 ProQuest Info&Learning
File  56:Computer and Information Systems Abstracts 1966-2007/Sep
         (c) 2007 CSA.
```

```
File  60:ANTE: Abstracts in New Tech & Engineer 1966-2007/Aug
         (c) 2007 CSA.
File  62:SPIN(R) 1975-2007/Sep W1
         (c) 2007 American Institute of Physics
File  65:Inside Conferences 1993-2007/Sep 27
         (c) 2007 BLDSC all rts. reserv.
File  95:TEME-Technology & Management 1989-2007/Sep W3
         (c) 2007 FIZ TECHNIK
File  99:Wilson Appl. Sci & Tech Abs 1983-2007/Aug
         (c) 2007 The HW Wilson Co.
File 111:TGG Natl.Newspaper Index(SM) 1979-2007/Sep 19
         (c) 2007 The Gale Group
File 144:Pascal 1973-2007/Sep W3
         (c) 2007 INIST/CNRS
File 239:Mathsci 1940-2007/Oct
         (c) 2007 American Mathematical Society
File 256:TecInfoSource 82-2007/May
         (c) 2007 Info.Sources Inc
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 2006 The Thomson Corp
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
```

17/7/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

07557029 INSPEC Abstract Number: B2000-05-1265D-028, C2000-05-5320G-017
Title: A low-voltage 42.4 G-BPS single-ended read- modify - write **bus and** programmable **page-** size **on a 3D frame-** buffer
Author(s): Inoue, K.; Abe, H.; Mori, K.; Fukagawa, S.
Author Affiliation: Syst.-LSI Div., Mitsubishi Electr. Corp., Itami, Japan
Journal: IEICE Transactions on Electronics vol.E83-C, no.2 p. 195-204
Publisher: Inst. Electron. Inf. & Commun. Eng,
Publication Date: Feb. 2000 Country of Publication: Japan
CODEN: IELEEJ ISSN: 0916-8524
SICI: 0916-8524(200002)E83C:2L.195:V4SE;1-8
Material Identity Number: P712-2000-002
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P); Experimental (X)
Abstract: Various kinds of high bandwidth architecture using embedded DRAM technology have been presented previously. In most cases, they use wide bus implementation and/or fast bus speed, which both have a die area penalty and a high power consumption penalty at the same time. The proposed single-ended read- **modify** - **write** bus doubles the bandwidth, while maintaining the **same** bus **size** and the **same** bus speed. The data-bus comprises a 1 kbit read-bus and a 1 kbit write-bus which work concurrently, with amplitude from 0 V to 1 V, and hence the measured power consumption is only 0.3 W at a frequency of 166 MHz. A **programmable** page- **size** reduces the page miss-rate and efficiently improves the bandwidth to be comparable to the wide bus and high speed approach. All the proposed features are implemented on a 3D frame- **buffer** to achieve 42.4 GBPS bandwidth. (7 Refs)
Subfile: B C

17/7/2 **(Item 1 from file: 8)**
DIALOG(R)File 8:Ei Compendex(R)

08548449 E.I. No: EIP00055156927
Title: Low-voltage 42.4 G-BPS single-ended read- modify - write **bus and**
programmable **page-** size **on a 3D frame-** buffer
Author: Inoue, Kazunari; Abe, Hideaki; Mori, Kaori; Fukagawa, Shuji
Corporate Source: Mitsubishi Electric Corp, Itami-shi, Jpn
Source: IEICE Transactions on Electronics v E83-C n 2 2000. p 195-204
Publication Year: 2000
CODEN: IELEEJ ISSN: 0916-8524
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 0006W4
Abstract: Various kinds of high bandwidth architecture using the embedded
DRAM technology have been presented previously. In most cases, they use wide bus implementation and/or fast bus speed, that both have the penalty
of die area and much power consumption at the same time. The proposing single-ended read- **modify - write** bus increases the bandwidth twice as high, while it maintains the **same** bus **size** and the **same** bus speed.
The data-bus comprises 1 k-bit read-bus and 1 k-bit write-bus that each works concurrently, and has amplitude from 0 V to 1 V, hence the measured
power consumption is only 0.3 W at a frequency of 166 MHz. A **programmable**
page- **size** reduces the page miss-rate and efficiently improves the bandwidth that is comparable to the wide bus and fast speed approach. All
the proposing features are implemented on a 3D frame- **buffer** to achieve
42.4 G-BPS bandwidth. (Author abstract) 7 Refs.

17/7/3 (Item 1 from file: 56)
DIALOG(R)File 56:Computer and Information Systems Abstracts

0000348194 IP ACCESSION NO: 456425
Low-voltage 42.4 G-BPS single-ended read- modify - write **bus and**
programmable **page-** size **on a 3D frame-** buffer

Inoue, Kazunari; Abe, Hideaki; Mori, Kaori; Fukagawa, Shuji
Mitsubishi Electric Corp, Itami-shi, Jpn

IEICE Transactions on Electronics, v E83-C, n 2, p 195-204, 2000
PUBLICATION DATE: 2000

PUBLISHER: Oxford University Press, Walton St., Oxford, OX2 6DP
COUNTRY OF PUBLICATION: UK
PUBLISHER URL: http://www.oup.co.uk

DOCUMENT TYPE: Journal Article
RECORD TYPE: Abstract
LANGUAGE: English
ISSN: 0916-8524
FILE SEGMENT: Computer & Information Systems Abstracts

ABSTRACT:
Various kinds of high bandwidth architecture using the embedded DRAM technology have been presented previously. In most cases, they use wide bus implementation and/or fast bus speed, that both have the penalty of die area and much power consumption at the same time. The proposing single-ended read- **modify - write** bus increases the bandwidth twice as high, while it maintains the **same** bus **size** and the **same** bus speed.
The data-bus comprises 1 k-bit read-bus and 1 k-bit write-bus that each works concurrently, and has amplitude from 0 V to 1 V, hence the measured power consumption is only 0.3 W at a frequency of 166 MHz. A **programmable** page- **size** reduces the page miss-rate and efficiently improves the bandwidth that is comparable to the wide bus and fast speed approach. All the proposing features are implemented on a 3D frame- **buffer** to achieve 42.4 G-BPS bandwidth.

17/7/4 (Item 1 from file: 144)
DIALOG(R)File 144:Pascal

14610441 PASCAL No.: 00-0279672
Low-voltage 42.4 G-BPS single-ended read- modify - write **bus and** programmable **page-** size **on a 3D frame-** buffer
INOUE K; ABE H; MORI K; FUKAGAWA S
Mitsubishi Electric Corp, Itami-shi, Japan
Journal: IEICE Transactions on Electronics, 2000, v E83-C (2) 195-204
ISSN: 0916-8524 Availability: INIST-26604
No. of Refs.: 7 Refs.
Document Type: P (Serial) ; A (Analytic)
Country of Publication: Japan
Language: English
Various kinds of high bandwidth architecture using the embedded DRAM technology have been presented previously. In most cases, they use wide bus implementation and/or fast bus speed, that both have the penalty of die area and much power consumption at the same time. The proposing single-ended read- **modify** - **write** bus increases the bandwidth twice as high, while it maintains the **same** bus **size** and the **same** bus speed. The data-bus comprises 1 k-bit read-bus and 1 k-bit write-bus that each works concurrently, and has amplitude from 0 V to 1 V, hence the measured power consumption is only 0.3 W at a frequency of 166 MHz. A **programmable** page- **size** reduces the page miss-rate and efficiently improves the bandwidth that is comparable to the wide bus and fast speed approach. All the proposing features are implemented on a 3D frame- **buffer** to achieve 42.4 G-BPS bandwidth.

20/7/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

09892486
Title: A nonredundant ternary CAM circuit for network search engines
Author(s): Akhbarizadeh, M.J.; Nourani, M.; Vijayasarathi, D.S.; Balsara, T.
Author Affiliation: Cisco Syst. Inc, San Jose, CA, USA
Journal: IEEE Transactions on Very Large Scale Integration (VLSI) Systems
vol.14, no.3 p.268-78
Publisher: IEEE,
Publication Date: March 2006 Country of Publication: USA
CODEN: IEVSE9 ISSN: 1063-8210
SICI: 1063-8210(200603)14:3L.268:NTCN;1-8
Material Identity Number: P986-2006-004
DOI: 10.1109/TVLSI.2006.871760
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P); Experimental (X)
Abstract: An optimized Ternary CAM concept is introduced for the hardware search engines in high-speed Internet routers. Our design employs w + 1 RAM bits to store a word of size w, whereas a conventional TCAM needs 2w RAM bits for the **same** word **size** . Based on this concept an 8-bit cluster is designed out of 9 SRAM bits, used as the basic building block of our **Prefix** -CAM (PCAM) structure. Four such clusters merge to **store a** 32-bit IPv4 **prefix** , thus, configuring a PCAM suitable for Internet packet forwarding. This PCAM module employs 48% less SRAM cells and a total of 22% less transistors plus 50% less address decode interconnects compared to a conventional TCAM, for **equal** storage **size** and **equal functionality** . We show that PCAM can be employed for multifield packet classification. Other factors, such as lookup speed and power dissipation, are not adversely affected. (31 Refs)
Subfile: B C

20/7/2   **(Item 2 from file: 2)**
DIALOG(R)File   2:INSPEC

09538538   INSPEC Abstract Number: C2005-09-6150C-052
Title: **Maintaining consistency and bounding capacity of software code caches**
Author(s): Bruening, D.; Amarasinghe, S.
Author Affiliation: Comput. Sci. & Artificial Intelligence Lab., MIT, Cambridge, MA, USA
Conference Title: International Symposium on Code Generation and Optimization   p.74-85
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA
Publication Date: 2005   Country of Publication: USA   xv+339 pp.
ISBN: 0 7695 2298 X   Material Identity Number: XX-2005-00553
U.S. Copyright Clearance Center Code: 0-7695-2298-X/05/$20.00
Conference Title: International Symposium on Code Generation and Optimization
Conference Date: 20-23 March 2005   Conference Location: San Jose, CA, USA
Language: English   Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Software **code** **caches** are becoming ubiquitous, in dynamic optimizers, runtime tool platforms, dynamic translators fast simulators and emulators, and dynamic compilers. **Caching** frequently executed fragments of **code** provides significant performance boosts, reducing the overhead of translation and emulation and meeting or exceeding native performance in dynamic optimizers. One disadvantage of **caching** , **memory** expansion, can sometimes be ignored when executing a single application. However, as optimizers and translators are applied more and more in production systems, the **memory** expansion from running multiple applications simultaneously becomes problematic. A second drawback to **caching** is the added **requirement** of maintaining consistency between the **code** **cache** and the original **code** . On architectures like IA-32 that do not require explicit application actions when **modifying** **code** , detecting **code** changes is challenging. Again, consistency can be ignored for certain sets of applications, but as **caching** systems scale up to executing large, modern,

complex **programs** , consistency becomes critical. This paper presents
efficient schemes for **keeping** a software **code** **cache** consistent and
for dynamically bounding **code** **cache** **size** to **match** the current
working set of the application. These schemes are evaluated **in** the
DynamoRIO runtime **code** manipulation system, and operate on stock hardware
in the presence of multiple threads and dynamic behavior, including
dynamically-loaded, generated, and even **modified** **code** . (37 Refs)
Subfile: C

20/7/3   **(Item 3 from file: 2)**
DIALOG(R)File   2:INSPEC

06483317   INSPEC Abstract Number: C9703-4240C-006

Title: Feasible time-optimal algorithms for Boolean functions **on exclusive-** write **parallel random-access machines**

Author(s): Dietzfelbinger, M.; Kutylowski, M.; Reischuk, R.
Author Affiliation: Fachbereich Inf., Dortmund Univ., Germany

Journal: SIAM Journal on Computing   vol.25, no.6   p.1196-230
Publisher: SIAM,
Publication Date: Dec. 1996 Country of Publication: USA
CODEN: SMJCAT ISSN: 0097-5397
SICI: 0097-5397(199612)25:6L.1196:FTOA;1-G
Material Identity Number: S171-97001
U.S. Copyright Clearance Center Code: 0097-5397/96/$2.00+0.15
Language: English   Document Type: Journal Paper (JP)
Treatment: Practical (P)

Abstract: It was shown some years ago that the computation time **for** many important Boolean **functions** of n arguments on concurrent-read exclusive-write parallel random-access machines (CREW PRAMs) of unlimited **size** is **at least** phi (n) approximately=0.72 log/sub 2/ n. On the other hand, it is known that every Boolean **function** of n arguments can be computed in phi (n)+1 steps on a CREW PRAM with n.2/sup n-1/ processors and **memory** cells. In the case of the OR of n bits, n processors and cells are sufficient. In this paper, it is shown that for many important **functions** , there are CREW PRAM algorithms that almost meet the lower bound in that they take phi (n)+o(log n) steps but use only a small number of processors and **memory** cells (in most cases, n). In addition, the cells only have to store binary words of bounded length (in most cases, length 1). We call such algorithms "feasible". The **functions** concerned include the following: the PARITY **function** and, more generally, all symmetric **functions** ; a large class of Boolean formulas; some **functions** over non-Boolean domains {0, ..., k-1} for small k, in particular, parallel- **prefix** sums; addition of n-bit numbers; and sorting n/l binary numbers of length l. Further, it is shown that Boolean circuits with fan-in 2, depth d, and size s can be evaluated by CREW PRAMs with fewer then s processors

in phi (2/sup d/)+o(d) approximately=0.72d+o(d) steps. For the
exclusive-read exclusive-write (EREW) PRAM model, a feasible algorithm is
described that computes PARITY of n bits in 0.86 log/sub 2/ n steps. (33 Refs)

Subfile: C

20/7/4 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

06063494 INSPEC Abstract Number: C9511-6140D-028
Title: Distributed data access in AC
Author(s): Carlson, W.W.; Draper, J.M.
Author Affiliation: IDA Supercomput. Res. Center, Bowie, MD, USA
Journal: SIGPLAN Notices Conference Title: SIGPLAN Not. (USA) vol.30, no.8 p.39-47
Publication Date: Aug. 1995 Country of Publication: USA
CODEN: SINODQ ISSN: 0362-1340
Conference Title: Fifth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP
Conference Sponsor: ACM
Conference Date: 19-21 July 1995 Conference Location: Santa Barbara, CA, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: We have **modified** the C language to support a **programming** model based on a shared address space with physically distributed **memory**. With this model, called AC, users can **write programs** in which the nodes of a massively parallel processor can access remote **memory** without message passing. AC provides support for distributed arrays as well as pointers to distributed data. Simple array references and pointer dereferencing are sufficient to generate low-overhead remote reads and writes. We have implemented these ideas in a compiler based on the GNU C compiler and targeted at Cray Research's T3D. Initial performance measurements show that AC generates **code** for remote accesses which is considerably faster than that of the native compiler for structures up to about 16 words in **size** and virtually **equivalent** for larger transfers.
(17 Refs)
Subfile: C

20/7/5 **(Item 1 from file: 8)**

10781407 E.I. No: EIP05519609523
Title: Memory allocation for **embedded systems with a compile-time-unknown scratch-pad size**
Author: Nguyen, Nghi; Dominguez, Angel; Barua, Rajeev
Corporate Source: Electrical and Computer Engineering Department
University of Maryland, College Park, MD 20742, United States
Conference Title: CASES 2005: International Conference on Compilers,
Architecture, and Synthesis for Embedded Systems
Conference Location: San Francisco, CA, United States Conference Date:
20050924-20050927
Sponsor: ACM SIGMICRO; IEEE TC-uARCH; ACM SIGBED
E.I. Conference No.: 66240
Source: CASES 2005: International Conference on Compilers, Architecture,
and Synthesis for Embedded Systems CASES 2005: International Conference on
Compilers, Architecture, and Synthesis for Embedded Systems 2005.
Publication Year: 2005
ISBN: 159593149X
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical)
Journal Announcement: 0512W5
Abstract: This paper presents the first **memory** allocation scheme for embedded systems having scratch-pad **memory** whose size is unknown at compile time. A scratch-pad **memory** (SPM) is a fast compiler-managed SRAM that replaces the hardware-managed **cache** . Its uses are motivated by its better real-time guarantees as compared to **cache** and by its significantly lower overheads in energy consumption, area and access time. Existing data allocation schemes for SPM all require that the SPM size be known at compile-time. Unfortunately, the resulting executable is tied to that size of SPM and is not portable to processor implementations having a different SPM size. Such portability would be valuable in situations where **programs** for an embedded system are not burned into the system at the time of manufacture, but rather are downloaded onto it during deployment, either using a network or portable media such as **memory** sticks. Such post-deployment **code updates** are common in distributed networks and in personal hand-held devices. The presence of different SPM **sizes** in different devices is **common** because of the evolution in VLSI

technology
across years. The result is that SPM cannot be used in such situations with downloaded **code** . To overcome this limitation, this work presents a
compiler method whose resulting executable is portable across SPMs of any
size. The executable at run-time places frequently used objects in SPM; it
considers **code** , global variables and stack variables for placement in
SPM. The allocation is decided by **modified** loader software before the
program is first run and once the SPM size can be discovered. The loader
then **modifies** the **program** binary based on the decided allocation. To
keep the overhead low, much of the pre-processing for the allocation is
done at compile-time. Results show that our benchmarks average a 36% speed
increase versus an all-DRAM allocation, while the optimal static allocation scheme, which knows the SPM size at compile-time and is thus an
un-achievable upper-bound, is only slightly faster (41% faster than all-DRAM). Results also show that the overhead from our embedded loader
averages about 1% in both **code** -size and run-time of our benchmarks.
 27 Refs.

20/7/6 **(Item 2 from file: 8)**
DIALOG(R)File 8:Ei Compendex(R)

10210987 E.I. No: EIP05028777794
Title: Logic-enhanced memory **for 3D graphics tile-based rasterizers**
Author: Crisu, D.; Cotofana, S.D.; Vassiliadis, S.; Liuha, P.
Conference Title: The 2004 47th Midwest Symposium on Circuits and Systems
- Conference Proceedings
Conference Location: Hiroshima, Japan Conference Date:
20040725-20040728
Sponsor: IEEE Circuits and Systems Society; Hiroshima University
E.I. Conference No.: 64127
Source: Midwest Symposium on Circuits and Systems The 2004 47th Midwest
Symposium on Circuits and Systems - Conference Proceedings v 2 2004. (IEEE
cat n 04CH37540)
Publication Year: 2004
CODEN: MSCSDL ISSN: 1548-3746
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical); X;
(Experimental)
Journal Announcement: 0501W3
Abstract: An efficient logic-enhanced **memory** architecture to accelerate
primitive traversal in 3D graphics tile-based rasterizers is presented.
The **memory** contains the **same** **number** of **bits** as the **number** of
pixels in the tile, and during rasterization time it is filled up in
several clock cycles by a systolic primitive scan-conversion subsystem
with the stencil of the primitive: ones are written for **memory** locations
that represent tile pixels covered by primitive, otherwise zeros are
stored. Once the shape of the primitive has been **coded** inside the **memory**
, the **memory** internal logic is capable of delivering, on request, up to
four hit positions (positions inside the primitive) per clock cycle to the
pixel processing pipelines, signaling when all the hit positions were
consumed. The logic-enhanced **memory** architecture presents the following
benefits: it handles "ghost" primitives efficiently, hit positions are
communicated in a spatial pattern that increases the hit ratio of texture
caches in pull texture architectures, and hit positions can always be
mapped to different **memory** banks in the Z- **buffer** or color- **buffer**
breaking the "read- **modify** - **write** " dependency associated with depth test
and color blending, thus allowing efficient pipelining. Hardware
implementation in a typical 0.18mum process technology for a QVGA 3D
graphics hardware accelerator with a tile size of 32 multiplied by 16
pixels has **indicated** that the **memory** can be clocked at 200MHz and

consumes an area of 120000mum**2. 9 Refs.

20/7/7 **(Item 1 from file: 34)**
DIALOG(R)File 34:SCISEARCH(R) CITED REF SCI

08473399 Genuine Article#: 289LN Number of References: 7
Title: A low-voltage 42.4 G-BPS single-ended read- modify - write **bus and**
programmable **page-sire on a 3D frame-** buffer
Author(s): Inoue K (REPRINT) ; Abe H; Mori K; Fukagawa S
Corporate Source: MITSUBISHI ELECTR CORP,SYST LSI DIV AS MEMORY/ITAMI/HYOGO
6648641/JAPAN/ (REPRINT)
Journal: IEICE TRANSACTIONS ON ELECTRONICS, 2000, VE83C, N2 (FEB), P195-204
ISSN: 0916-8524 Publication date: 20000200
Publisher: IEICE-INST ELECTRONICS INFORMATION COMMUNICATIONS ENG,
KIKAI-SHINKO-KAIKAN BLDG MINATO-KU SHIBAKOEN 3 CHOME, TOKYO 105, JAPAN
Language: English Document Type: ARTICLE
Abstract: Various kinds of high bandwidth architecture using the embedded DRAM technology have been presented previously. In most cases, they use wide bus implementation and/or fast bus speed, that both have the penalty of die area and much power consumption at the same time. The proposing single-ended read- **modify - write** bus increases the bandwidth twice as high, while it maintains the **same** bus **size** and the **same** bus speed. The data-bus comprises 1 k-bit read-bus and 1 k-bit write-bus that each works concurrently, and has amplitude from 0 V to 1 V, hence the measured power consumption is only 0.3 W at a frequency of 166 MHz: A **programmable** page-size reduces the page miss-rate and efficiently improves the bandwidth that is comparable to the wide bus and fast speed approach. All the proposing features are implemented on a 3D frame- **buffer** to achieve 42.4 G-BPS bandwidth.

20/7/8 **(Item 1 from file: 35)**
DIALOG(R)File 35:Dissertation Abs Online

02013329 ORDER NO: AADAA-I3128270
Modeling shape effects in nano magnetic materials with Web based micromagnetics
Author: Zhao, Zhidong
Degree: Ph.D.
Year: 2004
Corporate Source/Institution: University of New Orleans (0108)
Adviser: Scott L. Whittenburg
Source: VOLUME 65/04-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 1923. 169 PAGES

This research work focuses on the geometry and shape effects on submicron magnetic material. A web based micromagnetics **program** is **written** to model the hysteresis loop of nano magnetic samples with arbitrary geometry shapes and multiple magnetic materials.

Three material samples have been modeled with this **program** along with nano magnets with a variety of geometric shapes.

Shape anisotropy has been introduced to a permalloy ring by adding a cross-tie structure with various widths. The in-plane hysteresis loop and reversal behavior have no notable difference in direction parallel to the cross-tie, but greatly changed in perpendicular and diagonal directions. The switching field distribution is significantly reduced. The two distinct "onion" bit states of the **modified** ring elements are stabilized in the hysteresis in the diagonal direction The changes in the **modified** rings make them better candidates for Magnetic Random Access **Memory** elements.

Two Pac-Man elements, PM I and PM II, geometrically **modified** from disc and half disc respectively, are modeled. The PM I element undergoes a magnetic reversal through a two-stage mechanism that involves nucleation in the left and right middle areas followed by vortex core formation and vortex core motion in the lower middle area. The reversal process of the PM II element lacks the vortex core formation and motion stage. The switching field of the PM I and PM II elements are the same but the switching field distribution of the PM II elements is much narrower than that of the PM I element. Only the PM II element meets MRAM application requirements.

The thickness dependence of the magnetic properties of a core-shell structure has been studied. The nano particles have a cobalt core and a permalloy shell. The nano spheres are the **same** **size** but with various

shell thickness. Simulations reveal a multi-stage reversal process without
the formation of a Bloch wall for thin shell structure and smooth reversal
process with the formation and motion of a Bloch wall for thick-shell structure. Gradual transition of the hysteresis loop patterns has been observed.

20/7/9 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01203942 ORDER NO: NOT AVAILABLE FROM UNIVERSITY MICROFILMS INT'L.
VECTORIZED INTERPROCESSOR COMMUNICATION AND DATA MOVEMENT IN SHARED-
MEMORY
MULTIPROCESSORS (VECTORIZED MEMORY)
Author: PANDA, DHABALESWAR KUMAR
Degree: PH.D.
Year: 1991
Corporate Source/Institution: UNIVERSITY OF SOUTHERN CALIFORNIA
(0208)
Chairman: KAI HWANG
Source: VOLUME 52/09-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 4838.

Vectorized **memory** access schemes have been used traditionally in multiprocessors to enhance computational efficiency. However, applications requiring dense communication and data manipulation are unable to take advantage of these **memory** access schemes. In this thesis, we take a new approach to vectorized shared- **memory** access with an objective of implementing processor- **memory** data movement, **memory** -to- **memory** data manipulation, and processor-processor communication, all in vectorized manner.

This thesis has two major contributions. The first contribution lies in developing a novel vectorized **memory** access scheme to blend with interleaved **memory** organization. During vector data transfer between processor and interleaved **memory** system, this scheme allows data elements of a vector to be manipulated on-the-fly under **program** control. Using this scheme, we develop a new concept of atomic vector read- **modify** - **write** cycle and demonstrate parallel data manipulation with minimal overhead from processors. With two-dimensional interleaved **memory** organization, we demonstrate up to 75% savings in computational bandwidth in implementing matrix shifts and rotations. This scheme demonstrates potential to achieve concurrent computation and data manipulation.

The second contribution is in developing a new concept of **memory** -based vectorized interprocessor communication on multiprocessors with interleaved shared **memories** . We configure this shared- **memory** as a collection of vector mailboxes. With a suitable allocation of these mailboxes, we demonstrate that processors can exchange messages by vector **memory** -write and **memory** -read accesses. Similar to vectorizing computational steps, this approach allows communication steps of a parallel **program** to be vectorized. We present a communication vectorization scheme. This scheme vectorizes interprocessor communication steps of a distributed- **memory** multicomputer **programs** and implements them on a

shared- **memory** multiprocessor. Due to vector-oriented communication, such **program** conversion leads to a significant reduction in communication complexity. Three multiprocessor configurations are evaluated in their capabilities to support this vectorization. Communication complexities in these multiprocessors are compared with those of a hypercube system using circuit-switched message passing. For applications requiring all-to-all type of dense message patterns, communication complexity reduces by a factor of two to four when a hypercube system is compared with a shared- **memory** multiprocessor of the **same size** . (Copies available exclusively from Micrographics Department, Doheny Library, USC, Los Angeles, CA 90089-0182.)

20/7/10 **(Item 1 from file: 144)**
DIALOG(R)File 144:Pascal

17577490 PASCAL No.: 06-0165451
PCAM : A ternary CAM optimized for longest prefix **matching tasks**
ICCD 2004 : IEEE International Conference on Computer Design : VLSI in computers & processors : proceedings : 11-13 October, 2004, San Jose, CA
AKHBARIZADEH Mohammad J; NOURANI Mehrdad; VIJAYASARATHI Deepak S; BALSARA Poras T
Center for Integrated Circuits & Systems The University of Texas at Dallas, Richardson, TX 75083, United States
IEEE computer society, United States; IEEE Circuits and systems society, United States; IEEE. Electron Devices Society, United States
IEEE International Conference on Computer Design, 22 (San Jose CA USA) 2004-10-11
2004 6-11
Publisher: IEEE Computer Society, Los Alamitos CA
ISBN: 0-7695-2231-9 Availability: INIST-Y 38741; 354000138729580010
No. of Refs.: 15 ref.
Document Type: C (Conference Proceedings) ; A (Analytic)
Country of Publication: United States
Language: English
An optimized Ternary CAM concept is introduced for application in the longest **prefix** matching tasks of the Internet search engines. It employs w + 1 RAM bits for a word of size w. A conventional TCAM needs 2w RAM bits for the **same** word **size** . Based on this concept an 8 bit **Prefix** -CAM cluster is designed out of 9 SRAM bits, four of which merge to **store** a 32-bit IPv4 **prefix** . A complete **Prefix** -CAM module employs 22% less transistors than a conventional TCAM, for **equal** storage **size** and **equal** **functionality** . We confirm the 22% area saving by implementing the layouts for **Prefix** -CAM and TCAM words. Our design also reduces interconnect area by reducing address decode lines.

20/7/11 (Item 1 from file: 239)
DIALOG(R)File 239:Mathsci

02831945 MR 98j#73001
The mechanics and thermodynamics of continuous media.
Silhavy, Miroslav (Mathematical Institute, Czech Academy of Sciences (AVCR), 115 67 Prague, Czech Republic)
Corporate Source Codes: CZ-AOS
Publ: Springer-Verlag, Berlin,
1997, pp. xiv+504 pp. ISBN: 3-540-58378-5
Series: Texts and Monographs in Physics.
Price: $89.50.
Language: English Summary Language: English
Document Type: Book
Journal Announcement: 9705
Subfile: MR (Mathematical Reviews) AMS
Abstract Length: LONG (242 lines)
FEATURED REVIEW. \par\noindent Both in depth and in its encyclopedic coverage, this treatise by Silhavy evokes **memory** of the justly famous Handbuch articles of C. A. Truesdell, III and R. Toupin [in Handbuch der
Physik, Bd. III/1, 226--793; appendix, pp. 794--858, Springer, Berlin, 1960; MR 22\#8778] and Truesdell and W. Noll [The non-linear field theories of mechanics, Springer, Berlin, 1965; MR 33\#2030]. Indeed it may
be regarded as an ambitious attempt on the part of the author to write a
sequel to the Handbuch articles, covering a large portion of the major developments in continuum thermostatics and thermomechanics in the three
decades since Truesdell and Noll's account of their revival in the 1959--65
papers of Bernard D. Coleman and his coworkers.

Commenting on the axioms of continuum physics, Truesdell and Noll, in
their preface to The non-linear field theories [op. cit.], spoke about ``a
principle of irreversibility'', the ``true form'' of which ``is not yet known''. Search for the true form of this principle occupied the center stage in thermomechanics for the first half of the three decades since then. Miroslav Silhavy made his first impact as a researcher in thermomechanics by creating, concurrently with and independently of James.
B. Serrin, what is now sometimes called the Serrin-Silhavy approach to thermodynamics.

In the work of Coleman and his coworkers in the sixties, the entropy and
the absolute temperature were taken as given a priori, and the Clausius-Duhem inequality (revived by Truesdell and Toupin [op. cit.]) was
postulated as the principle of irreversibility or the Second Law in continuum thermomechanics. Perhaps arising partly as attempts to allay criticisms from proponents of other approaches, there soon followed a stream of foundational studies (the efforts of W. A. Day and of Coleman and
Owen being good representatives) examining the general validity of the

Clausius-Duhem inequality and investigating the existence and uniqueness of entropy, or the lack thereof, for various material bodies.

These efforts set the stage for the work of Serrin and of Silhavy in the late seventies and early eighties.

Following the pioneers of classical thermodynamics, Serrin and Silhavy take the concepts of hotness, heat and work as primitives. As was intuitively clear to Carnot and was explicitly pointed out by Gibbs, ``in thermodynamic problems, heat received at one temperature is by no means the equivalent of the same amount of heat received at another temperature$\ldots\,$. This is a result of the general law, that heat can only pass from a hotter to a colder body$\ldots\,$.'' In other words, we should consider in thermodynamics not only the quantity but also the quality of heat as characterized by the hotness at which the heat is received or given out. If a body receives a quantity of heat over a range of temperatures, just knowing the total amount received will not suffice for us to carry out a full thermodynamic analysis; in addition, the quality of the heat received must be specified by a detailed breakdown of the total according to the temperatures of receipt.

It will be a problem for future historians of thermodynamics to explain why a suitable mathematical expression to capture both the quantitative and qualitative aspects of heat was not created until Serrin and Silhavy, around 1978, independently proposed their `` **accumulation function** '' and ``heat distribution measure'', respectively, so that once and for all classical thermodynamics could be set on a firm mathematical foundation. A general theory of thermodynamics was subsequently developed. I would call this theory neoclassical thermodynamics, for it is best described as the classical thermodynamics of Clausius and Kelvin put in the most general setting (plus a careful delineation of the required axioms and logical arguments). In neoclassical thermodynamics, the existence of the mechanical equivalent of heat and that of an absolute temperature scale follow as general theorems. Furthermore, whenever the existence of a local entropy **function** can be demonstrated for a classical continuous body, the Clausius-Duhem inequality will indeed be the ``true form'' of the principle of irreversibility. For simple cases such as the thermoelastic solid and the heat-conducting Navier-Stokes fluid, it has been explicitly shown that

the entropy does exist as a local state **function** .

The theory of neoclassical thermodynamics is presented in Part II of the present treatise. It is a masterful account and is more comprehensive than either the outline by Serrin [in New perspectives in thermodynamics, 3--32, Springer, Berlin, 1986; see MR 87h:80002 \refcno848766\endrefcno] or the elementary exposition by D. R. Owen [A first course in the mathematical foundations of thermodynamics, Springer, New York, 1984; MR 85m:80001], although these three presentations each carry distinguishing personal touches of their authors and none of them can be regarded as complete. In particular, discussion of applications in Silhavy's book is mainly restricted to systems with perfect accessibility.

Occupying less than one-tenth of the page space notwithstanding, this part puts an unmistakable imprint on the rest of the book. It affects the overall organization and the selection of topics. First and foremost, the Clausius-Duhem inequality is taken throughout as the principle of irreversibility. Secondly, the constitutive equations treated in this book are restricted mainly to elastic materials with heat conduction and viscosity. Restricting attention to this special class of materials will soften any objection to taking the Clausius-Duhem inequality as the Second Law, for it is only the particular form of this law for the specific bodies in question that is at issue. For elastic materials with heat conduction and viscosity, most thermomechanical theories will have their principle of irreversibility equivalent to the Clausius-Duhem inequality. On the other hand, even within the Serrin-Silhavy approach itself, I am not aware of a published proof that such bodies in the most general case do obey the accessibility axiom, although hardly anyone would doubt the validity of this assertion for them.

The present treatise is certainly not meant to be a survey of thermomechanical theories. As for other approaches in thermomechanics, apart from a brief mention of Muller's entropy inequality (p. 165), three short sections ($\S\S 12.2$--12.4) in small print are devoted to linear irreversible thermodynamics, the dissipation potential, and relaxation models (extended linear irreversible thermodynamics), respectively, which are all ``completely compatible with the Clausius-Duhem inequality''.

Silhavy's treatise is divided into five parts. Like the article of Truesdell and Noll, it begins with a chapter on tensor algebra and analysis. This chapter together with Chapter 8 on isotropic **functions** in Part III gives a valuable compilation of formulae on tensor **functions**

many of which were discovered after the publication of Truesdell and Noll's Handbuch article. In Part I (entitled ``Balance equations''), besides the standard fare, there are well-written sections on Bravais lattices ($\S1.5$), compatibility of deformations at the interface ($\S2.3$), rank-1 connections ($\S2.4$) and twins ($\S2.5$), which will help prepare the reader for later studies of phase transitions in crystals. There are also appendices on piecewise smooth objects ($\S2.6$) and on the Gauss-Green theorem ($\S3.9$); a brief review of sets of finite perimeter and **functions** of bounded variation is given in the latter.

In Part II, ``Foundations'', there is also a chapter in which Silhavy gives an exposition of his own work (1989) to derive Cauchy's equations of motion from the balance of energy and the principle of material frame-indifference. In his approach mass is a derived concept, and the classical splitting of the total energy into the kinetic energy plus the objective internal energy is a consequence, not a presumption. Interestingly enough, it was also Serrin who, basing his ideas partly on those of Silhavy, came up with a related approach to these concepts at about the same time.

Part III is devoted to the constitutive equations of elastic materials with heat conduction and viscosity. Special cases include Navier-Stokes-Fourier fluids, Kelvin-Voigt solids, thermoelastic materials, and ideal dissipationless materials. The restrictions placed on the response **functions** by frame-indifference, symmetry, and the Clausius-Duhem inequality are derived.

Part IV, which runs to about two hundred pages long, treats the theory of thermodynamic equilibrium. It is a celebration of the point of view championed by J. L. Ericksen since the late sixties, namely that thermodynamics as taught by Gibbs is not only the theory of heat, but also a theory of equilibrium and stability, with the main tool the extremum principles and the calculus of variations. This part begins with chapters describing different types of environments, the equilibrium states of a body in a given environment, and the extremum principles. Then various notions of convexity (including quasiconvexity, rank-1 convexity, and polyconvexity) are presented in two chapters entitled ``Convexity'' and ``Constitutive inequalities'', respectively. There is a section ($\S17.4$) on Maxwell's relation, in which the continuity of the normal component of the Eshelby energy-momentum tensor across the static phase interface is established as the generalization of the equality of chemical potentials, which has been known since the days of Gibbs as a condition of equilibrium

for the special case of fluids. After ``Constitutive inequalities'' follow a chapter on the thermostatics of fluids, and one on the linearized approach to the equilibrium of solids, which includes classical linear elasticity and the linearized elasticity of stressed bodies. The final chapter of this part is devoted to direct methods in equilibrium theory. It begins with a section which describes the main mathematical ingredients, namely weak convergence, Young measures, and the lower semicontinuity of integral **functionals** . The analysis of Ball (1977) and some further developments on solutions to extremum problems for rubber-like bodies, which have polyconvex **stored** energy **functions** , are presented. The final section gives an exposition of the work of Chipot and Kinderlehrer (1988) on Young measure minimizers and the equilibrium configurations of crystals.

Part V, entitled ``Dynamics'', has its emphasis placed on moving singular surfaces, which include propagating phase boundaries and shock waves. Here the Clausius-Duhem inequality gives us the entropy admissibility criterion for jumps. This criterion, however, is not strong enough to secure uniqueness of solutions for the initial-value problem. In this part, extra conditions for evolving phase boundaries and various admissibility criteria for shocks are reviewed. Besides shock waves, various types of elastic waves (surface waves, acceleration waves, etc.) are also discussed, and there is a chapter devoted to adiabatic fluid dynamics (shock waves, shock layers). The book ends with a chapter in which the properties of a linearized system of equations for a viscous solid with heat conduction are examined.

The treatment given to the topics covered is encyclopedic. It includes accounts of much that is of secondary or even tertiary importance to the main theme of the book. Those accounts are often given in small print but are supported by a full bibliography.

While this treatise is impressive in its scope of coverage, it does have major omissions. Particularly notable among these are the gradient theory of phase transformations and M. E. Gurtin's theory of configurational forces and phase interfaces with structure (i.e., phase boundaries that are stressed and carry energy and entropy). Both of these topics are compatible with the general theme of this book; in fact, they are mentioned briefly in small print, and the main references are listed in the bibliography. On the other hand, exclusion of these topics from detailed discussion is also

understandable. All the topics treated in the present book fit together nicely within a relatively simple framework: the balance equations and the form of the Clausius-Duhem inequality are classical; the existence of internal energy and entropy as local state **functions** is not in doubt for the special classes of materials in question. In this sense the aforementioned omitted topics are outcasts; besides foundational issues on the existence and uniqueness of internal energy and entropy as local state **functions** , which remain to be clarified as the author should reexamine these issues from the standpoint of the Serrin-Silhavy approach, these theories will also require significant **modification** of the structure of the balance equations.

Most chapters of this treatise carry a bibliographical note, in which the author comments on the literature and provides information on the history of and other approaches to the subject. I find these bibliographical notes informative and helpful.

This book has an excellent bibliography, comprising twenty-two pages. The subject index, however, can be improved substantially. Since the whole treatise is roughly the **same** **size** as Truesdell and Noll's book [op. cit.], which has a subject index about three times as long, this shortcoming sometimes renders it difficult to locate quickly items of secondary or tertiary importance.

This treatise is clearly not directed to beginners but to scholars, specialists, and researchers who are already active in continuum thermomechanics, and to advanced students who desire to begin research on the subject. Its encyclopedic coverage, exhaustive bibliography, and helpful bibliographical notes make it a valuable book of reference. Supplemented by the original papers, this book can also serve as the basic reference or roadmap for several topic courses or seminars for advanced graduate students. I used it profitably in a seminar in which students learnt about various notions of convexity, Young measures, and the equilibrium theory of crystals and rubber-like bodies.

I strongly recommend this book to research libraries and to all practitioners of continuum thermomechanics.

Reviewer: Man, Chi-Sing (1-KY)

Review Type: Featured review

?
22/7/1 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

06723667 INSPEC Abstract Number: C9711-5220P-028
Title: Conflict-free access to templates of trees and hypercubes in parallel memory **systems**
Author(s): Das, S.K.; Pinotti, M.C.
Author Affiliation: Dept. of Comput. Sci., North Texas Univ., Denton, TX, USA
Conference Title: Computing and Combinatorics. Third Annual International Conference COCOON '97 Proceedings p.1-10
Editor(s): Jiang, T.; Lee, D.T.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 1997 Country of Publication: Germany xiii+522 pp.
ISBN: 3 540 63357 X Material Identity Number: XX97-01897
Conference Title: Proceedings of Third Annual International Computing and Combinatorics Conference
Conference Date: 20-22 Aug. 1997 Conference Location: Shanghai, China
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: We deal with the problem of mapping data structures, called hosts, into as few distinct **memory** modules as possible to guarantee that sets of distinct host nodes, called templates, can be accessed in parallel and without **memory** conflicts. An efficient solution to this important problem leads to a higher **memory** bandwidth and a better overall performance of a multiprocessor system. Considering a binomial tree as the host, we devise for the first time a recursive mapping of its nodes which allows conflict-free access to any binomial subtree. Since the overlappings among various template instances intricate the problem, thus requiring more **memory** modules than the template size, we define what are called the oriented templates (sub-trees) for which the conflict-freeness is guaranteed using the number of **memory** modules **equal** to the template **size** . We also investigate the conflict-free access to d-dimensional subcubes of n-dimensional hypercubes. In this context, hypercubes model sets of items indexed with n-digit (binary or non-binary) in which

parallel
accesses will be made to sets of items differing in an arbitrary collection
of d-digit positions. With the help of the coding theory, we propose a
novel approach to solve the subcube access problem. Codes with minimum
distance d>or=2 play a crucial role in our applications. In fact, we prove
that any occurrence of a subcube Q/sub s/ contained in/implied by Q/sub n/,
for 0<or=s<or=d-1, can be accessed without conflicts using [2/sup n//M]
memory modules, by associating an n-dimensional hypercube, Q/sub n/, with
a linear **code** G of length n, **size** M and minimum distance d. Associating
the hypercube nodes with maximum distance separable (MDS) codes, our
problem is solved optimally both in terms of the number of **memory** modules
required and the amount of load per module. These codes can be easily
modified (without node relocation) when the **size** of the host or the
number of available **memory** modules change. (9 Refs)

Subfile: C

22/7/2 **(Item 1 from file: 144)**
DIALOG(R)File 144:Pascal

13237785 PASCAL No.: 97-0507157
Conflict-free access to templates of trees and hypercubes in parallel memory systems
COCOON '97 : computing and combinatorics : Shanghai, August 20-22, 1997
DAS S K; PINOTTI M C
TAO JIANG, ed; LEE DT, ed
Dept of Computer Sciences, Univ of North texas, Denton, TX 76203, United States; IEI, Consiglio Nazionale delle Ricerche, Via S. Maria 46, 56126 Pisa, Italy
Annual international computing and combinatorics conference, 3 (Shanghai CHN) 1997-08-20
Journal: Lecture notes in computer science, 1997, 1276 1-10
ISBN: 3-540-63357-X ISSN: 0302-9743 Availability: INIST-16343; 354000061707580010
No. of Refs.: 9 ref.
Document Type: P (Serial); C (Conference Proceedings) ; A (Analytic)
Country of Publication: Germany; United States
Language: English
In this paper, we deal with the problem of mapping data structures, called hosts, into as few distinct **memory** modules as possible to guarantee that sets of distinct host nodes, called templates, can be accessed in parallel and without **memory** conflicts. An efficient solution to this important problem leads to a higher **memory** bandwidth and a better overall performance of a multiprocessor system. Considering a binomial tree as the host, we devise for the first time a recursive mapping of its nodes which allows conflict-free access to any binomial subtree. Since the overlappings among various template instances intricate the problem, thus requiring more **memory** modules than the template size, we define what are called the oriented templates (subtrees) for which the conflict-freeness is guaranteed using the number of **memory** modules **equal** to the template **size** . We also investigate the conflict-free access to d-dimensional subcubes of n-dimensional hypercubes. In this context, hypercubes model sets of items indexed with n-digit (binary or non-binary) in which parallel accesses will be made to sets of items differing in an arbitrary

collection
of d-digit positions. With the help of the coding theory, we propose a
novel approach to solve the subcube access problem. Codes with minimum
distance d >2 play a crucial role in our applications. In fact, we prove
that any occurrence of a subcube Q SUB s C Q SUB n , for 0 < s < d - 1, can
be accessed without conflicts using (2 SUP n /M) **memory** modules, by
associating an n-dimensional hypercube, Q SUB n , with a linear **code** C of
length n, **size** M and minimum distance d. Associating the hypercube nodes
with maximum distance separable (MDS) codes, our problem is solved
optimally both in terms of the number of **memory** modules required and the
amount of load per module. These codes can be easily **modified** (without
node relocation) when the **size** of the host or the number of available
memory modules change.

```
Set      Items   Description
S1     2732464   COMMAND? ? OR INSTRUCTION? ? OR PROGRAM? OR PROGRAMME?
? OR
                CODE? OR CODING? OR FUNCTION?
S2      302805   S1(5N)(MERG??? OR FUSE? ? OR FUSING OR UNIFY? OR
UNIFIE? ?
               OR UNITE? OR SYNTHESI? OR COMBIN? OR INTEGRAT? OR INCLU?
OR I-
               NCORPORAT?)
S3      503030   ENTEN? OR PREDICAT? OR PREFIX? OR PRE()(FIX???) OR
SUFFIX?
               OR MMX OR MODIF?
S4       21991   S2 AND S3
S5        3321   S4(5N)(STORE? ? OR STORING OR SAVE? ? OR ACCUMULAT? OR
KEE-
               P??? OR WRIT??? OR UPDAT?)
S6     1405157   BUFFER? OR MEMOR? OR CACHE? OR CACHING?? OR QUEUE?
S7       74586   S6(3N)(SPECIF? OR INDICAT? OR DESIR? OR REQUIR? OR
NECESS?
               OR CERTAIN? OR ACCEPT?)
S8      160990   (WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR
BITS))(5N)(EQUAL? -
               OR AT()LEAST OR COMPARABL? OR IDENTICAL? OR EQUIVALEN? OR
SAME
                OR SIMILAR)
S9       14841   (WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR
BITS))(5N)(MATCH? -
               OR AGREE? OR ALIKE OR AKIN OR CONGRUEN? OR COMMON? OR
INCOMMO-
               N?)
S10      29083   S1(5N)(WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR BITS))
S11       5723   S3(5N)(WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR BITS))
S12          8   S5 AND S6:S7 AND S8:S9 AND S10 AND S11
File 350:Derwent WPIX 1963-2007/UD=200761
         (c) 2007 The Thomson Corporation
File 347:JAPIO Dec 1976-2007/Jun(Updated 070926)
         (c) 2007 JPO & JAPIO
```

12/69,K/1 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0015575040 - Drawing available
WPI ACC NO: 2006-139202/200615
XRPX Acc No: N2006-120229
Method for accessing indirect memory in flash memory card involves, using native application program interface to mask memory access, Java fields as indirect memory elements or Java arrays as indirect memory elements
Patent Assignee: TEXAS INSTR FRANCE (TEXI); TEXAS INSTR INC (TEXI)
Inventor: BADL E; CABILLIC G; CHAUVEL G; D'INVERNO D; DINVERNO D; KUUSELA M
; LASSERRE S; LESOT J; MAJOUL S; MEQUIN J; PELTIER M
Patent Family (40 patents, 112 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 1622009	A1	20060201	EP 2004291918	A	20040727	200615	B
US 20060023517	A1	20060202	US 2005186062	A	20050721	200615	E
US 20060025986	A1	20060202	US 2005188310	A	20050725	200615	E
US 20060026126	A1	20060202	US 2005189245	A	20050726	200615	E
US 20060026183	A1	20060202	US 2005186063	A	20050721	200615	E
US 20060026200	A1	20060202	US 2005187199	A	20050722	200615	E
US 20060026201	A1	20060202	US 2005188550	A	20050725	200615	E
US 20060026312	A1	20060202	US 2005188667	A	20050725	200615	E
US 20060026322	A1	20060202	US 2005188923	A	20050725	200615	E
US 20060026353	A1	20060202	US 2005188491	A	20050725	200615	E
US 20060026354	A1	20060202	US 2005188668	A	20050725	200615	E
US 20060026357	A1	20060202	US 2005188411	A	20050725	200615	E
US 20060026370	A1	20060202	US 2005186271	A	20050721	200615	E
US 20060026390	A1	20060202	US 2005186315	A	20050721	200615	E
US 20060026391	A1	20060202	US 2005188827	A	20050725	200615	E
US 20060026392	A1	20060202	US 2005135796	A	20050524	200615	E
US 20060026393	A1	20060202	US 2005186239	A	20050721	200615	E
US 20060026394	A1	20060202	US 2005186330	A	20050721	200615	E
US 20060026395	A1	20060202	US 2005116522	A	20050428	200615	E
US 20060026396	A1	20060202	US 2005116893	A	20050428	200615	E
US 20060026397	A1	20060202	US 2005116897	A	20050428	200615	E
US 20060026398	A1	20060202	US 2005116918	A	20050428	200615	E
US 20060026400	A1	20060202	US 2005188311	A	20050725	200615	E
US 20060026401	A1	20060202	US 2005188336	A	20050725	200615	E
US 20060026402	A1	20060202	US 2005188503	A	20050725	200615	E
US 20060026403	A1	20060202	US 2005188592	A	20050725	200615	E
US 20060026404	A1	20060202	US 2005188502	A	20050725	200615	E
US 20060026405	A1	20060202	US 2005188504	A	20050725	200615	E
US 20060026407	A1	20060202	US 2005188309	A	20050725	200615	E
US 20060026412	A1	20060202	US 2005186036	A	20050721	200615	E
US 20060026563	A1	20060202	US 2005188551	A	20050725	200615	E
US 20060026564	A1	20060202	US 2005188670	A	20050725	200615	E
US 20060026565	A1	20060202	US 2005189422	A	20050726	200615	E
US 20060026566	A1	20060202	US 2005189637	A	20050726	200615	E
US 20060026571	A1	20060202	US 2005189367	A	20050726	200615	E
US 20060026574	A1	20060202	US 2005189211	A	20050726	200615	E
US 20060026575	A1	20060202	US 2005189410	A	20050726	200615	E
US 20060026580	A1	20060202	US 2005189411	A	20050726	200615	E

WO 2006127856 A2 20061130 WO 2006US20162 A 20060524 200680
NCE
US 7260682 B2 20070821 US 2005188668 A 20050725 200755
NCE

Priority Applications (no., kind, date): EP 2004291918 A 20040727; US 2005135796 A 20050524; US 2005188668 A 20050725; WO 2006US20162 A 20060524

Patent Details

Number Kind Lan Pg Dwg Filing Notes
EP 1622009 A1 EN 91 85
Regional Designated States,Original: AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IT LI LT LU LV MC MK NL PL PT RO SE SI SK TR
WO 2006127856 A2 EN
National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KM KN KP KR KZ LC LK LR LS LT LU LV LY MA MD MG MK MN MW MX MZ NA NG NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SM SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
Regional Designated States,Original: AT BE BG BW CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IS IT KE LS LT LU LV MC MW MZ NA NL OA PL PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW

Alerting Abstract EP A1

NOVELTY - The method involves using native application program interface (API) to mask the code sequences for **memory** access, Java fields as indirect **memory** elements or Java arrays to provide an abstraction of **memory** for contiguous set of **memory** elements.

USE - For accessing indirect **memory** in input/output device, flash **memory** card, non-addressable **memory** banks, etc., using Java language.

ADVANTAGE - Increases response time of the application and decreases energy consumption of the platform, while providing a simple technique to manage very specific in high level way.

DESCRIPTION OF DRAWINGS - The figure illustrates the process for accessing indirect **memory** .

Title Terms/Index Terms/Additional Words: METHOD; ACCESS; INDIRECT; **MEMORY** ; FLASH; CARD; NATIVE; APPLY; PROGRAM; INTERFACE; MASK; FIELD; ELEMENT; ARRAY

Class Codes

International Classification (+ Attributes)
IPC + Level Value Position Status Version

G06F-0012/00 A I F B 20060101
G06F-0013/24 A I F B 20060101
G06F-0013/28 A I F B 20060101
G06F-0017/00 A I F B 20060101
G06F-0017/30 A I F B 20060101
G06F-0007/00 A I F B 20060101
G06F-0009/00 A I F B 20060101
G06F-0009/30 A I F B 20060101
G06F-0009/34 A I F B 20060101
G06F-0009/40 A I L B 20060101
G06F-0009/44 A I F B 20060101
G06F-0009/45 A I F B 20060101
G06F-0009/455 A I F B 20060101
G11C-0007/10 A I F B 20060101
G06F-0013/00 A N L B 20060101
G06F-0012/00 C I L B 20060101
G06F-0013/20 C I F B 20060101
G06F-0017/00 C I L B 20060101
G06F-0017/30 C I L B 20060101
G06F-0007/00 C I L B 20060101
G06F-0009/00 C I L B 20060101
G06F-0009/30 C I L B 20060101
G06F-0009/34 C I F B 20060101
G06F-0009/40 C I L B 20060101
G06F-0009/44 C I L B 20060101
G06F-0009/45 C I L B 20060101
G06F-0009/455 C I L B 20060101
G11C-0007/10 C I L B 20060101
G06F-0012/00 C I B 20060101
G06F-0013/00 C N B 20060101

US Classification, Issued: 365189050, 703026000, 707002000, 707100000, 707103R00, 707103Y00, 710023000, 712228000, 710260000, 711118000, 711170000, 711133000, 711118000, 711170000, 711132000, 711154000, 712200000, 712209000, 712210000, 712210000, 712221000, 712221000, 712223000, 712223000, 712224000, 712224000, 712226000, 712221000, 712226000, 712226000, 712226000, 712227000, 712227000, 712228000, 712242000, 712242000, 717118000, 717118000, 717118000, 717147000, 717118000, 717147000, 717133000, 712227000, 717157000, 717140000, 717146000, 717140000, 717151000, 717151000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F03A; T01-H01B3A; T01-H05B2; T01-J20A; T01-J20B1

Method for accessing indirect memory in flash memory card involves, using native application program interface to mask memory access, Java fields as indirect memory elements or Java arrays as indirect memory elements

Original Titles:
...Emulating a direct **memory** access controller...

... **Memory** usable in **cache** mode or scratch pad mode to reduce the frequency of **memory** accesses...

... **Cache** **memory** usable as scratch pad storage...

...Context save and restore with a stack-based **memory** structure...
...Method and system for accessing indirect **memories**
...

... **Memory** access instruction with optional error check...

...Automatic operand load, **modify** and **store**
...

...Removing local RAM **size** limitations when executing software **code**
...

...Method and system for managing virtual **memory**

Alerting Abstract ...The method involves using native application program
interface (API) to mask the code sequences for **memory** access, Java fields
as indirect **memory** elements or Java arrays to provide an abstraction of
memory for contiguous set of **memory** elements.USE - For accessing
indirect **memory** in input/output device, flash **memory** card,
non-addressable **memory** banks, etc., using Java language...

...DESCRIPTION OF DRAWINGS - The figure illustrates the process for
accessing indirect **memory** .

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
...Methods, computer-readable media, and systems for dynamic address
translation between a source **memory** space and a target **memory** space are
provided. In some illustrative embodiments, a method is provided for
copying data from a source **memory** space to a target **memory** space. The
method includes extracting a plurality of source data units, each of size s
bits, from the source **memory** space and translating the plurality of
source data units into a plurality of target data units. A target data unit
is an addressable unit of the target **memory** space and each target data
unit is of size t bits. The method further includes...

...into a plurality of contiguous transfer units, each of size b bits, in
the target **memory** space...An electronic device comprising a first
processor adapted to process software instructions from a **memory** , and a
second processor coupled to the first processor. The second processor is

adapted to interrupt the first processor and to use the first processor as
a direct **memory** access (DMA) controller. The second processor uses the
first processor as a DMA controller by...

...instructions which, when executed, causes the first processor to load a
datum directly from a **memory** location and to transfer the datum to a
different **memory** location...

...A processor adapted to couple to external **memory** . The processor
comprises a controller and data storage (e.g., **cache** **memory**). The data
storage is configurable to operate in either a **cache** policy mode in which
a miss results in an access of the external **memory** or in a scratch pad
policy mode in which a miss does not result in an access of the external
memory . The data storage comprises a first portion and a second portion,
and only one of...

...A processor adapted to couple to external **memory** . The processor
comprises a controller and data storage. The data storage is usable to
store local variables and temporary data and is configurable to operate in
either a **cache** policy mode in which a miss results in an access of the
external **memory** or in a scratch pad policy mode in which a miss does not
result in an access of the external **memory** . The data storage comprises
first and second portions, and wherein only one of said portions...
...A multi-threaded processor adapted to couple to external **memory**
comprises a controller and data storage operated by the controller. The
data storage comprises a...

...second thread, only one of the first or second portions is cleaned to
the external **memory** if one of the first or second portions does not
contain valid data...

...Systems, methods, and storage media for accessing indirect **memory** in
Java applications are provided. In some embodiments, a storage medium is
provided that comprises Java application software that performs one or more
operations on an indirect **memory** of a device. The software comprises
instructions that create an instance of a Java class representing the
indirect **memory** , and instructions that access a **memory** element of the
indirect **memory** using an element unique identifier ("euid") of the
memory element. Other embodiments provide a method for accessing **memory**

elements of a device that comprises creating an instance of a Java class
representing the **memory** elements, and accessing a **memory** element of the
memory elements using an element unique identifier ("euid") of the **memory** element, wherein the **memory** elements are not mapped into the data
memory space of the processor...
...A method and system of informing a micro-sequence of operand **width**
.
At least some of the illustrative embodiments may be a method comprising
fetching a first opcode, asserting a flag if the first opcode **modifies** an
operand **width** of a subsequent opcode, fetching a second opcode,
triggering a micro-sequence based on the...

...to values in a second stack external to the core. The system also
comprises a **memory** coupled to the processor. In an iterative process, the
processor pops a data value off of the first stack and begins to store the
data value to the **memory** while the processor begins to use an existing
data value from the first stack to...the predetermined value, the load
instruction causes the processor to cause a data value from **memory** to be
loaded into a destination register...

...in the data structure, the decode logic obtains the operand from the
first storage unit, **modifies** the operand, and **stores** the operand to the
second storage unit for use by the group of instructions...

...A processor comprising fetch logic adapted to fetch instructions from
memory and decode logic coupled to the fetch logic and adapted to decode
the fetched instructions...A processor comprising fetch logic adapted to
fetch a set of instructions from **memory** , the set comprising a subset of
instructions. The processor further comprises decode logic coupled to...

...including an individual instruction and a first group of instructions.
The device further comprises a **memory** externally coupled to the
processor, as well as a second group of instructions. When executed...

...Methods, computer-readable media, and systems for virtual **memory**
management in Java(TM) are provided. In some illustrative embodiments, a
computer-readable medium storing a Java program that, when executed by a
processor, performs a method for virtual **memory** management is provided.

The method includes creating a Java representation of a page table, wherein
...

...storing a Java program that, when executed by a processor, performs a
method for virtual **memory** management that includes creating a Java representation of a segment descriptor, changing a field of...Some illustrative embodiments are a processor comprising fetch logic that retrieves an instruction from a **memory** , the instruction being part of a
program, and decode logic coupled to the fetch logic...

...methods and apparatus that fetch a first opcode, assert a flag if the
first opcode **modifies** an operand **width** of a subsequent opcode, fetch a
second opcode, trigger a micro-sequence based on the...
Claims:
...What is claimed is:1. A method for copying data from a source **memory** space to a target **memory** space, the method comprising:extracting
a plurality of source data units from the source **memory** space, wherein
each source data unit is of size s bits;translating the plurality of...

...target data units, wherein a target data unit is an addressable unit of
the target **memory** space and each target data unit is of size t bits; andcopying the plurality of target data units into a plurality of contiguous transfer units in the target **memory** space, wherein each transfer unit is of size b bits electronic device, comprising:a first processor adapted to process software instructions from a **memory** ; anda
second processor coupled to the first processor, said second processor adapted to interrupt the first processor and to use the first processor as
a direct **memory** access (DMA) controller;wherein the second processor uses
the first processor as a DMA controller...

...instructions which, when executed, causes the first processor to load a
datum directly from a **memory** location and to transfer the datum to a different **memory** location...

...What is claimed is:1. A processor adapted to couple to external
memory , comprising:a controller;data storage operated by said controller,
said data storage configurable to operate in either a **cache** policy mode
in which a miss results in an access of the external **memory** or in a scratch pad policy mode in which a miss does not result in an access of the
external **memory** ;wherein said data storage comprises a first portion and a

second portion, and wherein only...

...What is claimed is:1. A processor adapted to couple to external **memory** , comprising:a controller;data storage operated by said controller, said data storage usable to store local variables and temporary data and said data storage configurable to operate in either a **cache** policy mode in which a miss results in an access of the external **memory** or in a scratch pad policy mode in which a miss does not result in an access of the external **memory** ;wherein said data storage comprises a first portion and a second portion, and wherein only...

...What is claimed is:1. A multi-threaded processor adapted to couple to external **memory** , comprising:a controller;data storage operated by said controller, said data storage comprises a first...

...second thread, only one of said first or second portions is cleaned to the external **memory** if one of said first or second portions does not contain valid data...storage medium comprising Java application software that performs one or more operations on an indirect **memory** of a device, said software comprising:instructions that create an instance of a Java class representing the indirect **memory** ; andinstructions that access a **memory** element of the indirect **memory** using an element unique identifier ("euid") of the **memory** element...

...1. A method comprising:fetching a first opcode;asserting a flag if the first opcode **modifies** an operand **width** of a subsequent opcode;fetching a second opcode, and triggering a micro-sequence based on ...

...first stack corresponding to values in a second stack external to said core; anda **memory** coupled to the processor;wherein, in an iterative process, the processor pops a data value off of the first stack and begins to store the data value to said **memory** while the processor begins to use an existing data value from the first stack to ...plurality of registers coupled to the ALU;wherein, based on a control bit in a **memory** access instruction, said processor executes said instruction by causing contents of a source register to...

...value, said instruction causes said processor to cause a data value

to be moved between **memory** and a data register, said predetermined value being used to calculate a valid **memory** address from which to load the data value...

...in the data structure, the decode logic obtains the operand from the first storage unit, **modifies** the operand, and **stores** the operand to the second storage unit for use by said group of instructions...

...1. A processor, comprising:fetch logic adapted to fetch instructions from **memory** ; anddecode logic coupled to said fetch logic and adapted to decode said fetched instructions;wherein, if a bit... is:1. A processor, comprising:a fetch logic that retrieves a first instruction from a **memory** ;a decode logic coupled to the fetch logic; anda data structure at least partially within the **memory** ;wherein the decode logic decodes the first instruction and triggers execution of a first micro ...

...1. A processor, comprising:fetch logic adapted to fetch a set of instructions from **memory** , said set comprising a subset of instructions;decode logic coupled to the fetch logic and...

...comprising:a processor including an individual instruction and a first group of instructions; anda **memory** externally coupled to the processor and comprising a second group of instructions;wherein, when executed...

...storing a Java program that, when executed by a processor, performs a method for virtual **memory** management comprising:creating a Java representation of a page table, wherein each entry of the Java representation comp...is claimed is:1. A processor, comprising:fetch logic that retrieves an instruction from a **memory** , the instruction being part of a program; anddecode logic coupled to the fetch logic...

12/69,K/2 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0014662331 - Drawing available
WPI ACC NO: 2005-009912/200501
Related WPI Acc No: 2007-480868
XRPX Acc No: N2005-007778
Non-volatile memory **device e.g. programmable ROM, for computer system,**
has control unit to modify **information indicating** size **of boot** code
section upon receiving preset sequence of bus cycles to vary boot code
section size
Patent Assignee: ADVANCED MICRO DEVICES INC (ADMI)
Inventor: WISOR M T
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
US 6823435	B1	20041123	US 1997974971	A	19971120	200501	B

Priority Applications (no., kind, date): US 1997974971 A 19971120

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6823435	B1	EN	8	4	

Alerting Abstract US B1
NOVELTY - The device has a **memory** array (28) with a boot code section (36) that stores boot code. A storage unit stores information indicating **size** of the boot **code** section. A control unit (30) controls the storage and retrieval of data within and from the array, respectively. The **unit** (30) **modifies** the information indicating the **size** of the **code** section upon receiving a preset sequence of bus cycles to vary the **code** section **size** .

DESCRIPTION - An INDEPENDENT CLAIM is also included for a computer system with a non-volatile **memory** unit.

USE - Used for a computer system (claimed).

ADVANTAGE - The control unit **modifies** the information indicating the **size** of the boot **code** section upon receiving a preset sequence of bus cycles to vary the boot code section, thus enabling the usage of unused portion of the boot section to be utilized by system software for data storage and retrieval.

DESCRIPTION OF DRAWINGS - The drawing shows a block diagram of a flash **memory** unit.

20 **Memory** bus
28 **Memory** array
30 Control unit
32 Logic unit

36 Boot code section

Title Terms/Index Terms/Additional Words: NON; VOLATILE; **MEMORY** ;
DEVICE;
PROGRAM; ROM; COMPUTER; SYSTEM; CONTROL; UNIT; MODIFIED; INFORMATION;
INDICATE; SIZE; BOOT; CODE; SECTION; RECEIVE; PRESET; SEQUENCE; BUS;
CYCLE; VARY

Class Codes
International Classification (+ Attributes)
IPC + Level Value Position Status Version
G06F-0012/00 A I R 20060101
G06F-0012/00 C I R 20060101
US Classification, Issued: 711103000, 712037000, 713002000, 711170000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F01B; T01-F05B2

Non-volatile memory **device e.g. programmable ROM, for computer system,**
has control unit to modify **information indicating** size **of boot** code
section upon receiving preset sequence of bus cycles to vary boot code
section size

Original Titles:
Non-volatile **memory** system having a **programmably** selectable boot **code**
section **size**

Alerting Abstract ...NOVELTY - The device has a **memory** array (28) with
a boot code section (36) that stores boot code. A storage unit stores
information indicating **size** of the boot **code** section. A control unit
(30) controls the storage and retrieval of data within and from the array,
respectively. The unit (30) **modifies** the information indicating the **size**
of the **code** section upon receiving a preset sequence of bus cycles to
vary the **code** section **size** . DESCRIPTION - An INDEPENDENT CLAIM is also
included for a computer system with a non-volatile **memory** unit...

...ADVANTAGE - The control unit **modifies** the information indicating the
size of the boot **code** section upon receiving a preset sequence of bus
cycles to vary the boot code section...

...DESCRIPTION OF DRAWINGS - The drawing shows a block diagram of a flash
memory unit...

...20 **Memory** bus...

...28 **Memory** array...

Title Terms.../Index Terms/Additional Words: MEMORY ;

Original Publication Data by Authority

Original Abstracts:

A non-volatile **memory** system is presented having a boot **code** section, wherein the **size** of the boot **code** section may be programmably selected. One embodiment of the non-volatile **memory** system includes a **memory** array, a logic unit, a control unit, and a **program** **store** . The **memory** array **includes** multiple non-volatile **memory** cells (e.g., flash EEPROM cells). The **memory** array is divided into **memory** blocks of **equal** **size** . A number of the **memory** blocks are allocated for boot code storage, forming a boot code section of the **memory** array. The control unit controls storage of data within and retrieval of data from the **memory** array. The control unit includes a configuration register having a boot **code** section **size** field. The contents of the boot **code** section **size** field determine the number of **memory** blocks making up the boot code section. The logic unit is coupled between the control unit and the **memory** array, and receives address, data, and control signals from an external source. The logic unit provides the address, data, and control signals to the control unit and to the **memory** array. The program store stores instructions and data which determine the functionality of the control unit. Commands and configuration data are conveyed to the non-volatile **memory** system using predetermined sequences of bus write cycles. One embodiment of a computer system includes a central processing unit (CPU), and expansion bus, a **memory** bus, chip set logic, and the non-volatile **memory** system.

Claims:

What is claimed is: 1. The non-volatile **memory** device comprising: a **memory** array comprising a plurality of **memory** blocks, wherein a boot code section of the **memory** array is configured to store boot code; a storage unit, wherein a portion of the storage unit is configured to store information indicating a **size** of the boot **code** section; and a control unit configured to control storage of data within and retrieval of data from the **memory** array, wherein the control unit is further configured to vary the **size** of the boot **code** section by **modifying** the information indicating the **size** of the boot **code** section, wherein the control unit is further configured to modify the information in response to...

12/69,K/3 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0012340921 - Drawing available
WPI ACC NO: 2002-283110/200233
XRPX Acc No: N2002-221177
Calibration method for correcting synchronization errors in impulse widths in a device for testing an integrated circuit ensures that impulse widths do not adversely affect test measurements leading to false rejects
Patent Assignee: SCHLUMBERGER TECHNOLOGIES INC (SLMB)
Inventor: HELLAND J C
Patent Family (6 patents, 6 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
FR 2808333	A1	20011102	FR 20013388	A	20010313	200233	B
DE 10112311	A1	20020117	DE 10112311	A	20010314	200233	E
JP 2001305197	A	20011031	JP 200167243	A	20010309	200233	E
KR 2001092312	A	20011024	KR 200113100	A	20010314	200233	E
US 6496953	B1	20021217	US 2000526407	A	20000315	200307	E
TW 508446	A	20021101	TW 2001105048	A	20010409	200352	E

Priority Applications (no., kind, date): US 2000526407 A 20000315

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
FR 2808333	A1	FR	33	6	
JP 2001305197	A	JA	19		
TW 508446	A	ZH			

Alerting Abstract FR A1
NOVELTY - Method has the following steps: recording in **memory** , associated with a selected terminal of the device under test, of synchronization event data relating to the DUT, supply of function **data** (72) relating to the test, determining if the data cause a transition state in the DUT, with the state causing an impulse, adjustment of the synchronization event data so that the synchronization events are **matched** to the impulse **width** and generation of a test signal to apply to the DUT including the synchronization event adjusted for impulse width.
DESCRIPTION - The invention also relates to a calibration device for sending signals to a DUT for correcting synchronization errors in impulse widths.
USE - Device for ensuring that impulse widths sent from automatic test equipment to an integrated circuit (DUT) are correct and do not cause an erroneous error signal.
ADVANTAGE - In high performance automatic test devices for testing integrated circuits output signals can be adversely affected if impulse

widths of test signals applied to the DUT are not correct. The invention
provides a method for ensuring that impulse widths of signals used during
device testing are correct.

DESCRIPTION OF DRAWINGS - (Drawing includes non-English language text).
Figure shows a block diagram of the invention.

118DUT
70register assembly
72function data source
74decoder
102event sequence register.

Title Terms/Index Terms/Additional Words: CALIBRATE; METHOD; CORRECT; SYNCHRONISATION; ERROR; IMPULSE; WIDTH; DEVICE; TEST; INTEGRATE; CIRCUIT; ENSURE; ADVERSE; AFFECT; MEASURE; LEADING; FALSE; REJECT

Class Codes

International Classification (Main): G01R-031/3183, G06F-011/00
(Additional/Secondary): G01R-031/28, G01R-035/00, G11C-029/00
US Classification, Issued: 714744000, 714731000

File Segment: EPI;
DWPI Class: S01; U11; U22
Manual Codes (EPI/S-X): S01-G01A1; S01-G01A5; U11-F01C3; U22-H

...NOVELTY - Method has the following steps: recording in **memory** , associated with a selected terminal of the device under test, of synchronization event data relating...

...causing an impulse, adjustment of the synchronization event data so that
the synchronization events are **matched** to the impulse **width** and generation of a test signal to apply to the DUT including the synchronization event...

Original Publication Data by Authority

Original Abstracts:

...error during testing of an integrated circuit are described. The method
includes storing in a **memory** , associated with a selected terminal of an
integrated circuit, **event** timing data pertaining to testing of the integrated circuit. Functional data is provided, pertaining to...

...then applied to the selected terminal of the integrated circuit, the test signal including pulse **width** adjusted event timing.A test **program**
first loads scrambler **and** sequencer **memories** with a **code** representing
event timing data and event type data for a number **of** events that **are** to
occur during a test vector, as specified by the user. According to one

embodiment, to implement single value pulse **width** calibration, additional **coding** is provided that reflects variations on event timing values compensating for **pulse width** timing **error** . Circuitry external to the local event sequencer of the tester analyzes the **stream** of **functional** data describing event polarity during every test cycle, and determines if a given bit of...

...pulse. The results of this analysis become part of the data stored in the scrambler **memory** . These data act as a pointer to select the address in the sequencer **memory** **that** contains the correct pulse width adjusted event timing data. According to another embodiment **which** implements general pulse **width** calibration, **the** event sequencer is **modified** to include pulse **width** calculation circuitry, which stores **event** time and event type data **for** the most recent **events** and calculates the pulse width of the present event by subtracting the nominal time value...

Claims:

...timing errors for testing an integrated circuit, comprising the acts of:(a) storing in a **memory** , associated with a selected terminal of said integrated circuit, event timing data pertaining to testing of said **integrated** circuit;(b) providing **functional** data pertaining **to** said testing;(c) determining if said functional data causes a state transition in said integrated...

...state transition creating a pulse;(d) adjusting said event timing data, thereby to produce pulse **width** adjusted event **timing** ; and(e) generating a test signal to be applied to said selected terminal, said test...

12/69,K/4 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0009524387 - Drawing available
WPI ACC NO: 1999-468482/199939
XRPX Acc No: N1999-349804
Automatic window resizing method in graphical user interface
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: AMRO H Y
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
US 5940077	A	19990817	US 1996626197	A	19960329	199939	B

Priority Applications (no., kind, date): US 1996626197 A 19960329

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5940077	A	EN	7	3	

Alerting Abstract US A
NOVELTY - When the third resultant obtained is 30-70 % of window height, it is assigned as zoomed out size with respect to height. If the third resultant is 30-70 % of window height, the window height is reduced by 70 % and 30 % respectively. The **same** procedure is repeated for the **width** of window. If the window is in the zoomed out size, it is enlarged to default size, automatically.

DESCRIPTION - Based on the input command from the user, a display on a portion of window is controlled, to determine whether the window is in a zoomed outsize. When the window is not in the zoomed out size, height and width of window and computer display, are determined. Then the window height is squared and the squared value is divided by the height of computer display, to obtain a first resultant. Then division of 1' by total number of displayed windows, is carried out and then 1' is added to the division result, to obtain second resultant. The first and second resultants are then multiplied, to obtain third resultant. INDEPENDENT CLAIMs are also included for the following:

1.recording medium storing computer readable program for automatic window size modification ;

2.automatic window resizing system for automatically resizing window displayed in default state.

USE - For automatic window resizing in graphical user interface of operating systems such as windows (TM), OS/2 (TM) and AIX (TM)

operating
system.
ADVANTAGE - The contents displayed in a zoomed out window are proportionally reduced according to the amount of size reduction of the window, therefore only the entire window is reduced in size and the contents are retained in the original size. The GUI displays the new window
based on the default size and passes focus to it as in the case of newly
opened window.
DESCRIPTION OF DRAWINGS - The figure depicts the GUI display window for
displaying default window along with focus and zoomed out windows.

Title Terms/Index Terms/Additional Words: AUTOMATIC; WINDOW; METHOD; GRAPHICAL; USER; INTERFACE

Class Codes
International Classification (Main): G06F-015/00
US Classification, Issued: 345342000, 345340000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-J

Original Titles:
Method, **memory** and apparatus for automatically resizing a window while
continuing to display information therein.

Alerting Abstract ...70 % of window height, the window height is reduced
by 70 % and 30 % respectively. The **same** procedure is repeated for the **width** of window. If the window is in the zoomed out size, it is enlarged
to...
...recording medium storing computer readable **program** for automatic window **size** modification ; automatic window **resizing** **system** for automatically resizing window displayed in default state...

12/69,K/5 (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0009058714 - Drawing available
WPI ACC NO: 1998-112433/199811
XRPX Acc No: N1998-090094
Smart card with integrated circuit with processor and memory - has tickets with fields for storing entitlement data for ticket, validity data,
and data for checking ticket validity, and stores data using code with
fixed number of bits per bit group
Patent Assignee: KONINK KPN NV (NEPO); KONINK PTT NEDERLAND NV (NEPO)
Inventor: DRUPSTEEN M M P; MULLER F
Patent Family (11 patents, 32 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 823694	A1	19980211	EP 1996202240	A	19960809	199811	B
WO 1998007120	A1	19980219	WO 1997EP4333	A	19970807	199814	E
AU 199741180	A	19980306	AU 199741180	A	19970807	199830	E
EP 920681	A1	19990609	EP 1997938893	A	19970807	199927	E
			WO 1997EP4333	A	19970807		
AU 718123	B	20000406	AU 199741180	A	19970807	200027	E
US 6119945	A	20000919	US 1997908716	A	19970808	200048	E
NZ 334055	A	20010223	NZ 334055	A	19970807	200115	E
			WO 1997EP4333	A	19970807		
EP 920681	B1	20020220	EP 1997938893	A	19970807	200214	E
			WO 1997EP4333	A	19970807		
DE 69710588	E	20020328	DE 69710588	A	19970807	200229	E
			EP 1997938893	A	19970807		
			WO 1997EP4333	A	19970807		
ES 2172809	T3	20021001	EP 1997938893	A	19970807	200275	E
CA 2262760	C	20021105	CA 2262760	A	19970807	200281	E
			WO 1997EP4333	A	19970807		

Priority Applications (no., kind, date): EP 1996202240 A 19960809

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 823694	A1	EN	13	8	

Regional Designated States,Original: AT BE CH DE DK ES FI FR GB GR IE IT
LI LU MC NL PT SE

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 1998007120	A1	EN	23	8	

National Designated States,Original: AU CA CN CZ EE HU IL JP LT LV NO NZ
PL SG SI
Regional Designated States,Original: AT BE CH DE DK EA ES FI FR GB GR IE
IT LU MC NL PT SE

Number	Kind	Lan	Pg	Dwg	Filing Notes
AU 199741180	A	EN			Based on OPI patent WO 1998007120
EP 920681	A1	EN			PCT Application WO 1997EP4333
					Based on OPI patent WO 1998007120

Regional Designated States,Original: AT BE CH DE DK ES FI FR GB GR IE IT
LI LU NL PT SE

AU 718123 9741180	B	EN	Previously issued patent AU
			Based on OPI patent WO 1998007120
NZ 334055	A	EN	PCT Application WO 1997EP4333
			Based on OPI patent WO 1998007120
EP 920681	B1	EN	PCT Application WO 1997EP4333
			Based on OPI patent WO 1998007120

Regional Designated States,Original: AT BE CH DE DK ES FI FR GB GR IE IT LI LU NL PT SE

DE 69710588	E	DE	Application EP 1997938893
			PCT Application WO 1997EP4333
			Based on OPI patent EP 920681
			Based on OPI patent WO 1998007120
ES 2172809	T3	ES	Application EP 1997938893
			Based on OPI patent EP 920681
CA 2262760	C	EN	PCT Application WO 1997EP4333
			Based on OPI patent WO 1998007120

Alerting Abstract EP A1

The smart card comprises an integrated circuit with a processor having a **memory** . The **memory** is structured to comprises tickets (20). A ticket comprises an entitlement field (21) for storing data relating to the entitlement of the ticket.

A ticket further comprises a validation field (22) for storing data relating to the validity of the ticket, and a verification field (23) for storing data relating to a check of the validity of the ticket. Data is stored using a **code** containing a fixed **number** of set **bits** per group of bits. The **code** words have eight bits, and the set bits in each code word equal four.

USE - For tickets stored in smart cards and for using stored tickets. E.g. for electronic purse. Also for loyalty card or points card used by shops. Also for personal data such as medical record.

ADVANTAGE - Allows use of open tickets that is tickets which have non-predetermined validity date or time. Tickets can be securely stored.

Title Terms/Index Terms/Additional Words: SMART; CARD; INTEGRATE; CIRCUIT; PROCESSOR; **MEMORY** ; TICKET; FIELD; STORAGE; DATA; VALID; CHECK; CODE; FIX; NUMBER; BIT; PER; GROUP

Class Codes

International Classification (Main): G06K-019/06, G06K-019/07, G07F-007/08

(Additional/Secondary): G06K-019/073, G07B-015/00

US Classification, Issued: 235492000, 235380000, 235383000

File Segment: EPI;

DWPI Class: T01; T05

Manual Codes (EPI/S-X): T01-H01C1; T01-J05A; T05-C03; T05-H02C5C

Smart card with integrated circuit with processor and memory - ...

...**data for ticket, validity data, and data for checking ticket validity,**
and stores data using code **with fixed** number **of** bits **per bit group**

Alerting Abstract ...The smart card comprises an integrated circuit with
a processor having a **memory** . The **memory** is structured to comprises tickets (20). A ticket comprises an entitlement field (21) for storing...

...relating to a check of the validity of the ticket. Data is stored using
a **code** containing a fixed **number** of set **bits** per group of bits. The
code words have eight bits, and the set bits in each code word equal four
...

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
...a different, second storage command (WRITE) when validating the ticket.
Preferably, in the tickets a **code** is used having a fixed **number of** set
bits , **thus** preventing the **fraudulent modification** of an issued **ticket** .

...

...and using a different, second storage command (WRITE) when validating
the ticket. Thus the fraudulent **modification of** an issued ticket is prevented...

...a different, second storage command (WRITE) when validating the ticket
(20) at the validation terminal (82). Thus, the fraudulent **modification**
of an issued ticket (20) is prevented...
...and using a different, second storage command (WRITE) when validating
the ticket. Thus the fraudulent **modification** of an issued ticket is **prevented**.. >
Claims:
1. Smart card (1) comprising an integrated circuit (10) having a processor
(11) and a **memory** (12), the **memory** being **structured** so as to comprise tickets (20), a ticket comprising an entitlement field (21) for
storing...

...A smart card (1) comprising an integrated circuit (10) having a processor (11) and a **memory** (12), the **memory** being organized so as to comprise tickets (20), **a** ticket comprising **at** least one field (21; 22; . 23) for storing data relating to the ticket, characterized by...

...12) for storing data in the at least one field in the form of a **code** containing a fixed **number** of set **bits** per group **of** **bits** .

...

...operational exclusively in response to an identification of a **first** type of terminal, anda **memory** , **including** , **a** ticket **stored** in the **memory** having at least one field for storing data relating to the ticket;wherein the smart card is configured for storing data using a **code** containing a fixed **number** of **set** **bits** per group of bits.

12/69,K/6 (Item 6 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0007249957 - Drawing available
WPI ACC NO: 1995-303798/199540
XRPX Acc No: N1995-230816
Digital signal coding method for image sequence of blocks or macroblocks -
calculating image complexity and modifying quantisation step as function
of complexity with number of bits per group estimated according to complexity, number of images, bit rate and image frequency
Patent Assignee: LAB ELECTRONIQUE PHILIPS (PHIG); PHILIPS ELECTRONICS NV
(PHIG); PHILIPS GLOEILAMPENFAB NV (PHIG); US PHILIPS CORP (PHIG);
KONINK PHILIPS ELECTRONICS NV (PHIG)
Inventor: TRANCHARD L
Patent Family (5 patents, 7 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 670663	A1	19950906	EP 1995200384	A	19950217	199540	B
FR 2717029	A1	19950908	FR 19942382	A	19940302	199541	E
JP 7284109	A	19951027	JP 199540348	A	19950228	199601	E
US 5680483	A	19971021	US 1995392632	A	19950222	199748	E
JP 3818679	B2	20060906	JP 199540348	A	19950228	200659	E

Priority Applications (no., kind, date): FR 19942382 A 19940302; FR 19946380 A 19940526

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 670663	A1	FR	30	11	
Regional Designated States,Original: DE FR GB IT SE					
JP 7284109	A	JA	19	1	
US 5680483	A	EN	20	11	
JP 3818679	B2	JA	28		Previously issued patent JP 07284109

Alerting Abstract EP A1
The method involves defining the complexity of the image as the number of bits per image proportional to the **number of bits** issued during the **coding** step. The regulation stage includes evaluating the number of bits in each group, which is proportional to the complexity, the number of images, the bit rate at the coding output and the image frequency.

For each new image, the number of bits per image is estimated. The estimated value is then corrected (400), using limiting maximum and minimum values determined by the state of a **buffer memory** (10). For each macroblock, a coefficient of modification for a quantisation step is then calculated (110).

USE/ADVANTAGE - E.g. digital compression of video signal compatible with

MPEG-2 standard. Bit rate switching without broadcast interruption or interference allows increased flexibility.

Title Terms/Index Terms/Additional Words: DIGITAL; SIGNAL; CODE; METHOD;
IMAGE; SEQUENCE; BLOCK; CALCULATE; COMPLEX; MODIFIED; QUANTUM; STEP; FUNCTION; NUMBER; BIT; PER; GROUP; ESTIMATE; ACCORD; RATE; FREQUENCY

Class Codes
International Classification (Main): H04N-007/32
(Additional/Secondary): H04N-011/04, H04N-005/92
International Classification (+ Attributes)

IPC + Level	Value	Position	Status	Version
G06T-0009/00	A	I		R 20060101
H04N-0007/24	A	I		R 20060101
H04N-0007/32	A	I		R 20060101
H04N-0007/50	A	I		R 20060101
H04N-0007/60	A	I		R 20060101
H04N-0011/04	A	I	L	B 20060101
H04N-0005/92	A	I	L	B 20060101
H04N-0007/32	A	I	F	B 20060101
G06T-0009/00	C	I		R 20060101
H04N-0007/24	C	I		R 20060101
H04N-0007/32	C	I		R 20060101
H04N-0007/50	C	I		R 20060101
H04N-0007/52	C	I		R 20060101

US Classification, Issued: 348405000, 348419000, 382236000, 382251000, 382239000

File Segment: EPI;
DWPI Class: W02; W04
Manual Codes (EPI/S-X): W02-F07B; W02-F07C; W04-P01A3; W04-P01A5

...**calculating image complexity and modifying quantisation step as** function **of complexity with** number of bits **per group estimated according** to complexity, number of images, bit rate and image frequency

Alerting Abstract ...the complexity of the image as the number of bits
per image proportional to the **number** of **bits** issued during the **coding**
step. The regulation stage includes evaluating the number of bits in each
group, which is...

...then corrected (400), using limiting maximum and minimum values determined by the state of a **buffer memory** (10). For each macroblock, a
coefficient of modification for a quantisation step is then calculated...

Original Publication Data by Authority

Original Abstracts:
...the blocks or macroblocks, and a bitrate control sub-assembly. The sub-assembly includes a **buffer memory** and a **device for**

modifying
the quantization step used in quantization of the blocks or macroblocks.
The **buffer memory** and the **device for** modifying the quantization step are connected in series. The device for modifying the quantization step...
Claims:
...coded as a function of the complexity value thus computed, wherein:(A)
said complexity value **is** defined as a **number of bits** per picture **proportional** to the **number of bits** observed at the end of **coding** ;(B)
the step of controlling the bitrate comprising the following sub-steps:(1)
for each given group **of** N **successive** pictures, evaluating a **number** of
bits which comprise a group profile, said group profile having a value
PROF which is proportional to said complexity value, to the number N, and
to a bitrate R (t) **observed** at the **coding** output, and inversely proportional to the period of the pictures;(2) for each new picture **to** be
coded in the sequence:(a) estimating a **number of bits** corresponding to
said new picture and having a value NBNP which is proportional to **said** complexity value, and, a **number of bits per** sliding group having a
value NBSG which is proportional to said group profile value PROF...

...NBNP and two limit values MIN(CN) and MAX(CN) of the fullness of a **buffer memory** for **storing coded** signals, the selection criterion being the selection of that value which is between the other...

...values; and,(3) for each macroblock of said new picture, computation of
a coefficient for **modifying the** quantization **step** of said macroblock,
said coefficient being equal or proportional to the sum of a number of bits
expressing the initial fullness of a **buffer memory** defined as virtual
memory and a complementary number which **is equal** to the **number** of
bits already generated by **coding** the (j-1) macroblocks preceding the macroblock concerned of the rank j in the **same** picture, reduced **by** a
number of **correction bits** related to **the** values of j and CNPP **and**
to the number of macroblocks **per** picture.

12/69,K/7 (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0007022343 - Drawing available
WPI ACC NO: 1995-038009/199506
XRPX Acc No: N1995-030071
Image processing system with variable width memory bus for MPEG images -
has controller that splits word in half and stores each half separately,
and reassembles halves on read operation.
Patent Assignee: SGS THOMSON MICROELTRN SA (SGSA); STMICROELECTRONICS SA (SGSA)
Inventor: ALAIN A; ARTIERI A
Patent Family (7 patents, 6 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 632388	A1	19950104	EP 1994410044	A	19940627	199506	B
FR 2707118	A1	19950106	FR 19938218	A	19930630	199507	E
JP 7154781	A	19950616	JP 1994170329	A	19940630	199533	E
US 5825372	A	19981020	US 1994267195	A	19940629	199849	E
EP 632388	B1	19991222	EP 1994410044	A	19940627	200004	E
DE 69422228	E	20000127	DE 69422228	A	19940627	200012	E
			EP 1994410044	A	19940627		
JP 3787847	B2	20060621	JP 1994170329	A	19940630	200643	E

Priority Applications (no., kind, date): FR 19938218 A 19930630

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 632388	A1	FR	26	12	
Regional Designated States,Original: DE FR GB IT					
JP 7154781	A	JA	19		
EP 632388	B1	FR			
Regional Designated States,Original: DE FR GB IT					
DE 69422228	E	DE			Application EP 1994410044 Based on OPI patent EP 632388
JP 3787847	B2	JA	23		Previously issued patent JP 07154781

Alerting Abstract EP A1
The image processor co-operates with a **memory** that can store three decoded images. The **memory** is accessed by a bus N bits wide. For two processing modes a bus width of N/2 is used.
For each instruction to write an N-bit word to **memory** a circuit (62)
causes the successive writing of the first and second half of the word, using N/2 bits. To read **memory** the two half words are extracted and juxtaposed to re- form the N-bit word. An address controller generates two
distinct addresses for each address supplied by the processor. The clock
signal to the processor is inhibited to ensure each **memory** access instruction is executed twice.
USE/ADVANTAGE - MPEG decoder. Adapts to differing bus **widths**

without
modification of architecture of peripherals communicating with **memory**
,
and simplifies implementation of decoder.

Title Terms/Index Terms/Additional Words: IMAGE; PROCESS; SYSTEM;
VARIABLE;
WIDTH; **MEMORY** ; BUS; CONTROL; SPLIT; WORD; HALF; STORAGE; SEPARATE;
HALVES; READ; OPERATE; MPEG; DECODER

Class Codes
International Classification (Main): G06F-012/02, G06F-012/04, H04N-007/24
(Additional/Secondary): G06F-013/40
International Classification (+ Attributes)
IPC + Level Value Position Status Version

G06F-0012/04	A	I		R	20060101
G06F-0013/40	A	I		R	20060101
G06F-0009/34	A	I	L	B	20060101
H04N-0007/26	A	I		R	20060101
H04N-0007/26	A	I	F	B	20060101
H04N-0007/36	A	I		R	20060101
H04N-0007/50	A	I		R	20060101
G06F-0012/04	C	I		R	20060101
G06F-0013/40	C	I		R	20060101
H04N-0007/26	C	I		R	20060101
H04N-0007/36	C	I		R	20060101
H04N-0007/50	C	I		R	20060101

US Classification, Issued: 345512000, 345509000

File Segment: EPI;
DWPI Class: T01; W04
Manual Codes (EPI/S-X): T01-H01A; T01-H07A1; T01-J10A2; W04-P01A3;
W04-P01A5; W04-P01C5

Image processing system with variable width memory **bus for MPEG images**...

Original Titles:
...Processor system particularly for image processing comprising a variable
scize **memory** bus...

...Processor system particularly for image processing comprising a variable
size **memory** bus...

...IMAGE PROCESSING SYSTEM HAVING VARIABLE LENGTH **MEMORY** BUS...

...Image processing system including a variable size **memory** bus.

Alerting Abstract ...The image processor co-operates with a **memory** that
can store three decoded images. The **memory** is accessed by a bus N bits
wide. For two processing modes a bus width...

...For each instruction to write an N-bit word to **memory** a circuit (62) causes the successive writing of the first and second half of the word, using N/2 bits. To read **memory** the two half words are extracted and juxtaposed to re- form the N-bit word...

...supplied by the processor. The clock signal to the processor is inhibited to ensure each **memory** access instruction is executed twice...

...USE/ADVANTAGE - MPEG decoder. Adapts to differing bus **widths** without **modification** of architecture of peripherals communicating with **memory** , and simplifies implementation of decoder.

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
...comprising a data bus with a fixed size of N bits (D64) connected **to** a **memory** (12) for words **of** n bits by a bus with a size of n bits (D16), where N is...

...execution by the processor of an instruction for writing an N-bit word into the **memory** , successively writing each **sub** -word of n bits constituting this N-bit word to distinct addresses; and means (60...

...for, on each execution of an instruction for reading an N-bit word from the **memory** , successively reading from **the** said **memory** at distinct addresses **sub** -words of n bits, and juxtaposing these sub-words on the bus of fixed size...

...a data bus having a fixed N-bits size connected to an n-bits word **memory** through a bus having an n-bits size, **where** N is a multiple **of** n, and n is a variable value. The system includes means for, at each execution by the processor **of** a **write** **instruction** of one word of N bits **in** the **memory** , successively writing each sub-word of n bits **constituting** this word of N bits at distinct addresses, and means for, at each execution of a read instruction of a word of N bits in the **memory** , successively reading in this **memory** at distinct addresses **sub** -words of n bits, **and** juxtaposing these sub-words on the fixed size bus.
Claims:
...adapted to processing pictures according to intra, predicted and

bidirectional modes in cooperation with a **memory** (12) capable of storing at least three decoded pictures and accessible through an N-bit data bus, characterized in that **it** comprises, for processing pictures only according to the intra and predicted modes in cooperation with a half-size **memory** through an N/2-bit bus: - means (62) for, at each execution by the processor of a write **instruction of one** N-bit word in the **memory** , successively writing each N/2-bit sub-word constituting said N-bit word; - means (60, 64, 65, 66) for, at each **execution** of a read instruction of an N/bit word in the **memory** , successively reading in said **memory** two N/2-bit sub-words, and juxtaposing these sub-words on the N-bit bus; - an **address** folding circuit (52) comprising: - **an** addressing circuit (86) for providing the **memory** with two distinct addresses for each address provided by the processor; - an address generating circuit (80-84) for providing an address within the address **boundaries** of the **memory** when an address provided by the addressing circuit is out of predetermined boundaries; and - means (92, 94) for inhibiting the decoding if an address provided **to** the **memory** in write mode corresponds to data which has not yet been read.

...

...adapted to process images according to intra, predicted and bidirectional modes, in cooperation with a **memory** capable of storing at least three decoded images and accessible through an N-bit data bus, and adapted to process images only according to intra and predicted modes in cooperation with a half- **size memory through** an N/2-bit bus, said image processing system comprising: means for, at each execution by the processor of a write instruction of one N-bit word to the **half -size memory** , successively writing each N/2-bit sub-word of the N-bit word; means for instruction of an N-bit word from the half-size **memory** , **successively** reading in said half-size **memory** two N/2-bit sub-words, and juxtaposing these subwords on the N-bit bus; an addressing **circuit** for providing the half- **size memory** with two distinct **addresses for** each address provided by the **processor** ; an address folding circuit for providing an address within the address boundaries of the half-size **memory** when an address provided by the addressing means **is** out of the boundaries; and means for stopping the processor if an address provided

to
the half-size **memory** in write mode corresponds to data which has **not**
yet
been read.

12/69,K/8 (Item 8 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0003553510
WPI ACC NO: 1985-160661/198527
Related WPI Acc No: 1985-277861; 1990-255633; 2000-239401
High speed graphic pattern processing appts. - uses raster scan CRT and control unit with microprogram memory and decoder
Patent Assignee: HITACHI ENG CO LTD (HITJ); HITACHI LTD (HITA); HITACHI MFG CO (HITA); HITACHI SEISAKUSHO KK (HITA); KAJIWARA H (KAJI-I); KATSURA K (KATS-I); MAEJIMA H (MAEJ-I)
Inventor: KAJIWARA H; KATSURA K; MAEJIMA H
Patent Family (20 patents, 5 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 146961	A	19850703	EP 1984116285	A	19841224	198527	B
US 4862150	A	19890829	US 1984686039	A	19841224	198944	E
EP 146961	B	19910320	EP 1984116285	A	19841224	199112	E
DE 3484297	G	19910425				199118	E
US 5043713	A	19910827	US 1989350254	A	19890511	199137	E
US 5300947	A	19940405	US 1984686039	A	19841224	199413	E
			US 1989350254	A	19890511		
			US 1991737398	A	19910729		
US 5332995	A	19940726	US 1984686039	A	19841224	199429	E
			US 1989350254	A	19890511		
			US 1991736780	A	19910729		
KR 199410224	B1	19941022	KR 19848375	A	19841226	199638	E
			KR 199319698	A	19930925		
KR 199410225	B1	19941022	KR 19848375	A	19841226	199638	E
			KR 199319695	A	19930925		
KR 199507531	B1	19950711	KR 19848375	A	19841226	199715	E
			KR 199319696	A	19930925		
KR 199507532	B1	19950711	KR 19848375	A	19841226	199715	E
			KR 199319697	A	19930925		
US 5631668	A	19970520	US 1984686039	A	19841224	199726	E
			US 1989350254	A	19890511		
			US 1991736786	A	19910729		
			US 1993104572	A	19930811		
			US 1995430853	A	19950428		
US 5631671	A	19970520	US 1984686039	A	19841224	199726	E
			US 1989350254	A	19890511		
			US 1991736786	A	19910729		
			US 1993104572	A	19930811		
US 5638095	A	19970610	US 1984686039	A	19841224	199729	E
			US 1989350254	A	19890511		
			US 1991736786	A	19910729		
			US 1993104572	A	19930811		
			US 1995430848	A	19950428		
US 5657045	A	19970812	US 1984686039	A	19841224	199738	E
			US 1989350254	A	19890511		
			US 1991736786	A	19910729		
			US 1993104572	A	19930811		
			US 1995430851	A	19950428		
KR 199512931	B1	19951023	KR 199419887	A	19940812	199851	E
KR 199513229	B1	19951026	KR 19848375	A	19841226	199901	E

			KR 199419886	A	19940812		
KR 199707247	B1	00000000	KR 19848375	A	19841226	199941	E
US 20010052903	A1	20011220	US 1984686039	A	19841224	200206	E
			US 1989350254	A	19890511		
			US 1991736786	A	19910729		
			US 1993104572	A	19930811		
			US 1995430851	A	19950428		
			US 1997796983	A	19970207		
			US 1998161463	A	19980928		
			US 2001932895	A	20010821		
US 6492992	B2	20021210	US 1984686039	A	19841224	200301	E
			US 1989350254	A	19890511		
			US 1991736786	A	19910729		
			US 1993104572	A	19930811		
			US 1995430851	A	19950428		
			US 1997796983	A	19970207		
			US 1998161463	A	19980928		
			US 2001932895	A	20010821		

Priority Applications (no., kind, date): JP 1984120679 A 19840614; JP 198427155 A 19840217; JP 1983246986 A 19831226; JP 1984254889 A 19841130

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 146961	A	EN	70	28	
Regional Designated States,Original: DE FR GB IT					
EP 146961	B	EN			
Regional Designated States,Original: DE FR GB IT					
US 5300947	A	EN	36	28	Division of application US 1984686039
					Division of application US 1989350254
					Division of patent US 4862150
					Division of patent US 5043713
US 5332995	A	EN	33	28	Division of application US 1984686039
					Division of application US 1989350254
					Division of patent US 4862150
					Division of patent US 5043713
KR 199410224	B1	KO			Division of application KR 19848375
KR 199410225	B1	KO			Division of application KR 19848375
KR 199507531	B1	KO			Division of application KR 19848375
KR 199507532	B1	KO			Division of application KR 19848375

US 5631668 A EN 38 28 Division of application US
1984686039

Division of application US
1989350254

Division of application US
1991736786

Continuation of application US
1993104572

Division of patent US 4862150
Division of patent US 5043713
US 5631671 A EN 36 28 Division of application US
1984686039

Division of application US
1989350254

Division of application US
1991736786

Division of patent US 4862150
Division of patent US 5043713
US 5638095 A EN 34 28 Division of application US
1984686039

Division of application US
1989350254

Division of application US
1991736786

Continuation of application US
1993104572

Division of patent US 4862150
Division of patent US 5043713
US 5657045 A EN 34 28 Division of application US
1984686039

Division of application US
1989350254

Division of application US
1991736786

Continuation of application US
1993104572

Division of patent US 4862150
Division of patent US 5043713
KR 199513229 B1 KO Division of application KR
19848375

US 20010052903 A1 EN Division of application US
1984686039

Division of application US
1989350254

Division of application US 1991736786

Continuation of application US 1993104572

Continuation of application US 1995430851

Continuation of application US 1997796983

Continuation of application US 1998161463

Division of patent US 4862150
Division of patent US 5043713
Continuation of patent US 5631671
Continuation of patent US 5657045

US 6492992 B2 EN Division of application US 1984686039

Division of application US 1989350254

Division of application US 1991736786

Continuation of application US 1993104572

Continuation of application US 1995430851

Continuation of application US 1997796983

Continuation of application US 1998161463

Division of patent US 4862150
Division of patent US 5043713
Continuation of patent US 5631671
Continuation of patent US 5657045

Alerting Abstract EP A

The processing appts. can update one-pixed data, translate a logical address to physical address and transfer data in a display **memory** (13) at high speed. It comprises an operation unit (30) including logical address, physical address and colour data operation units, as well as a control unit (20).

The operation unit controls writing, updating and reading of display data. The control unit controls the operation unit in a predetermined sequence, the read display data is converted to a video signal by a conversion unit (40) for a display unit (50).

USE/ADVANTAGE - Drawing a pattern of multi-colour or multi-tone data having each pixel represented by a **number** of **bits** at the **same** processing speed as that for binary image data.

Equivalent Alerting Abstract US A

The appts. uses a raster scan type CRT. The graphic pattern

processing
appts. can update one pixel data, translate a logical address to a physical
address and transfer data in a display **memory** at high speed.

The graphic pattern processing appts. comprises an operation unit; including a logical address operation unit which stores and operates on logical coordinates, a physical address operation unit, which stores the
physical address of the **memory** corresponding to the current drawing point
and constants to perform an arithmetic operation to output a modified address to **memory** , a colour data operation unit and a control unit including microprogram **memory** and a microprogram decoder.

ADVANTAGE - High speed. (29pp)n

Equivalent Alerting Abstract US A

A graphic data processing apparatus is disclosed for accessing a **memory**
which stores pixels having a number of bits which may be selected. Graphic
data is generated with one or more bits per pixel with a number of pixels
of data being stored in one word of the **memory** .

A physical address operation unit stores information of a current drawing
point including a **memory** address of a word in the **memory** and a pixel
address defining a position of a pixel in one word **specified** by the **memory** address. A data operation unit **modifies** a particular pixel having
a **number** of **bits** which may be selected in the one word specified by the
pixel address in accordance with a drawing instruction.

ADVANTAGE - High processing speed. @(33pp)@

Equivalent Alerting Abstract US A

The graphic data generating appts. includes an information output device
for outputting an image of graphic data. A **memory** stores pixels of graphic data to be provided to the information output device. A pattern **memory** stores pattern data having at least one bit per pixel of the graphic data. A graphic data processor has a number of data storage elements, each storing pixel data having at least one bit.

A selector chooses one of the data storage elements depending upon a value of the at least one bit per of the graphic data, and a write device
writes the pixel data in the selected storage element as graphic data into
an address of the **memory** device.

ADVANTAGE - High speed calculation of address in display **memory** for **memory** update of multi-colour or multi-tone data.

Equivalent Alerting Abstract US A

The graphic data generator outputs graphic data of several pixels of an
image. A display **memory** is connected to the output for storing pixel data

defining the graphic data for each of the pixels.
Each pixel data has several bits. A graphic data processor performs readout of word data having several pixel data at a word position of the
display **memory** specified by a source **memory** address, selects pixel
data specified by a source pixel address in the readout word data, and writes the selected pixel data in the display **memory** at a pixel position
specified by a destination pixel address of word data which is **specified**
by a destination **memory** address.
USE/ADVANTAGE - Draws pattern of multi-colour or multi-tone data having
multi-bit pixels at same speed as binary image data.

Title Terms/Index Terms/Additional Words: HIGH; SPEED; GRAPHIC; PATTERN;
PROCESS; APPARATUS; RASTER; SCAN; CRT; CONTROL; UNIT; MICROPROGRAM;
MEMORY ; DECODE

Class Codes
International Classification (Main): G06F-012/10, G06F-015/62, G06K-009/00,
G06K-009/36, G06T-011/40, G09G-005/02, G09G-005/36, G09G-005/38
(Additional/Secondary): G06F-015/72, G06K-009/20, G09G-001/28, G09G-005/00
US Classification, Issued: 345552000, 345588000, 340703000, 340724000,
340744000, 340798000, 340799000, 395141000, 395166000, 340747000,
340744000, 395165000, 345155000, 345151000, 345121000, 345191000,
345155000, 345200000, 345133000, 345203000, 345028000, 345121000,
345191000, 345568000, 345501000, 345559000, 345565000

File Segment: EngPI; EPI;
DWPI Class: T01; T04; W02; W03; W04; P85
Manual Codes (EPI/S-X): T01-C04A; T01-J04; T04-H01B; W02-J03; W03-A04;
W04-P

...uses raster scan CRT and control unit with microprogram memory **and decoder**

Alerting Abstract ...pixed data, translate a logical address to physical
address and transfer data in a display **memory** (13) at high speed. It comprises an operation unit (30) including logical address, physical address...

...a pattern of multi-colour or multi-tone data having each pixel represented by a **number of bits** at the **same** processing speed as that
for binary image data.

Equivalent Alerting Abstract ...data, translate a logical address to a
physical address and transfer data in a display **memory** at high speed...

...on logical coordinates, a physical address operation unit, which stores
the physical address of the **memory** corresponding to the current drawing
point and constants to perform an arithmetic operation to output a modified
address to **memory** , a colour data operation unit and a control unit
including microprogram **memory** and a microprogram decoder...
...A graphic data processing apparatus is disclosed for accessing a **memory**
which stores pixels having a number of bits which may be selected. Graphic
data is...

...pixel with a number of pixels of data being stored in one word of the
memory .
...

...A physical address operation unit stores information of a current
drawing point including a **memory** address of a word in the **memory** and a
pixel address defining a position of a pixel in one word **specified** by the
memory address. A data operation unit **modifies** a particular pixel
having a **number** of **bits** which may be selected in the one word specified
by the pixel address in accordance...

...generating appts. includes an information output device for outputting
an image of graphic data. A **memory** stores pixels of graphic data to be
provided to the information output device. A pattern **memory** stores
pattern data having at least one bit per pixel of the graphic data. A...

...pixel data in the selected storage element as graphic data into an
address of the **memory** device...
...ADVANTAGE - High speed calculation of address in display **memory** for
memory update of multi-colour or multi-tone data...

...The graphic data generator outputs graphic data of several pixels of an
image. A display **memory** is connected to the output for storing pixel data
defining the graphic data for each...

...readout of word data having several pixel data at a word position of the
display **memory** **specified** by a source **memory** address, selects pixel
data specified by a source pixel address in the readout word data, and
writes the selected pixel data in the display **memory** at a pixel position
specified by a destination pixel address of word data which is

specified
by a destination **memory** address

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
...pixel data, translate a logical address to physical address and transfer
data in a display **memory** (13), at a high **speed** . The graphic pattern
processing apparatus comprises an operation unit (30) including logical address operation unit...

...320) and color data operation unit (330), and a control unit (20) including a microprogram **memory** (100) and a microprogram decoder (**200**).

...

...pixel data, translate a logical address to physical address and transfer
data in a display **memory** , at a high speed. The graphic pattern processing
apparatus comprises **an** operation unit including logical address operation
unit, physical address operation unit and color data operation unit, and a
control unit including a microprogram **memory** and a microprogram decoder.

...

...data, translate a logical address to a physical address and transfer
data in a display **memory** , at a high speed. The graphic pattern processing
apparatus comprises an operation unit including a logical address operation
unit, a **physical** address operation unit, color data operation unit, and a
control unit including a microprogram **memory** and a microprogram decoder.

...

...A graphic data processing apparatus is disclosed for accessing a **memory**
which stores pixels having a number of bits which may be selected. Gaphic
data is generated with one or more bits per pixel with a **plurality** of
pixels of data being stored in one word of the **memory** . A physical address
operation unit stores information of a current drawing point including a
memory address of a word in the **memory** and a pixel **address** defining a

position of a pixel in one word **specified** by the **memory** address. **A** data operation unit **modifies** a particular **pixel** having a **number** of **bits** which may be selected in the one word **specified** by the **pixel** address in accordance with a **drawing** **instruction** .

...

...A graphic data processing apparatus for accessing **memory** which stores pixels where each of the pixels is a picture element of a unique point in two-dimensional space and having a number of pixels which may be selected **in** the **memory** and for generating graphic data with two or more bits per pixel being used and a plurality of pixels of data being stored in one word of the **memory** is **disclosed** . A physical address operation unit stores information of a current drawing point including a **memory** address of a word in the **memory** and a pixel address defining a **position** of a pixel in one word **specified** by the **memory** address. A data operation unit modifies **a** particular pixel in the one word **specified** by the pixel address in accordance with a drawing instruction with a number **of** pixels within **a** word being selectable...

...apparatus includes an output producing a graphic image having a plurality of bits; a display **memory** connected to the output for storing pixel data defining the graphic data for each of...

...plurality of bits; and a graphic data processing apparatus performing read out of word data **having** a plurality of pixel data at a word position of the display **memory** **specified** by a source **memory** address, selecting pixel data specified by a source pixel address in the readout word and writing the selected pixel data in the display **memory** at a pixel position **specified** by **a** **destination** pixel address of **word** data **specified** by the **destination** **memory** address.

A graphic pattern processing apparatus having a graphic **memory** , a data processor, and a graphic processor. The graphic **memory** stores a pattern composed of pixel data. The graphics processor includes a plurality of color registers. The graphic processor reads the graphic **memory** in response to instructions received from the **data** processor. The graphics

processor in response to the pixel **data** read from the graphic **memory** selects one of a plurality of color registers and outputs that value.

A graphic pattern processing apparatus for accessing a **memory** which
stores words of graphic data. A plurality of pixels is stored in each word
...

...selected by a pixel address supplied by a graphic data processor. The
graphic data processor **performs** processing on the selected pixel in accordance with instructions received from a data processor...

...A graphic pattern processing apparatus having a display **memory**, a data
processor, a graphic processor, and a plurality of parallel to serial convertors. The display **memory** stores graphic data in words, each word
has a plurality of pixel data and each pixel data has a plurality of bits.
A graphic processor accesses the display **memory** and processes a plurality
of the pixel data in response to instructions received from a data processor. The number of parallel **to** serial convertors corresponds to the
number of bits per pixel and are configured to allow a word from the display **memory** to be converted into a serial stream **of** pixel data...
... A graphic data generating apparatus includes a data processor, a graphic **memory**, and a graphic processor. The data processor outputs instructions to the graphic processor for processing graphic data. The **instructions** include a drawing **instruction** for transferring graphic
data **stored** in a predetermined location in the graphic **memory** to another predetermined location in the graphic **memory**. The graphic **memory**
stores pixel data defining the graphic data and **each** of the pixel data
having a plurality of bits. The graphic processor performing read out of
word data **having** a plurality of **pixel** data at a word **position** of
the graphic **memory** **specified** by a **source** **memory** address, selecting
pixel data **specified** by **a** source pixel **address** in the readout word
and writing the selected pixel data in the graphic **memory** at a pixel position specified by a destination pixel address of word data **specified0**
by the destination **memory** address.

A data processing apparatus **which** processes data held in **memory**. The data processing apparatus includes an address operation unit
which obtains an address to read one-word data from the **memory**, wherein

the one-word data is a unit of data access to the **memory** , and a logical operation unit which determines a content of an operation on a field basis based on information which designates the number of bits **per** field to construct one-word data with a plurality of fields having a **same number** of **bits** . The logical operation **unit** , based on the content thus determined, performs the operation in parallel on the **fields** of the one-word data read from the **memory** by the address thus obtained.

Claims:

...pixed data, translate a logical address to physical address and transfer data in a display **memory** (13) at high speed. It comprises an operation unit (30) including logical address, physical address...

...1. A graphic pattern processing apparatus for use in connection with a display **memory** (13) for storing a display data therein and **means** (40) for converting the display data read from said display **memory** to a display signal for displaying on a **display** unit or transmission to other unit, the display data stored in the display **memory** being updated or processed for drawing in accordance **with** a **display** control data **including** an **instruction** and a parameter sent from a computer, characterised in that said graphic pattern processing apparatus...

...means (20) for generating an operation control signal to an operation unit and including microprogram **memory** means (100) for storing therein a microprogram for display control, **a** decoder (200) for decoding a microinstruction read from said microprogram **memory** and instruction control means (230) for controlling said decoder in **accordance** with the display control data; and an operation unit (30) including logical address operation means...

...display control means, physical address operation means (320) for calculating an address in said display **memory** based on the logical address and colour data operation means (330) **for** logically operating a selected multi-tone information or multi-colour information and the display data...

...1. In a graphic pattern processing apparatus having a display **memory** for storing a display data therein and means for converting the display data read from said display **memory** to a display signal for displaying on a display unit or transmission to other unit, the display **data** stored in the display **memory** being updated or processed for drawing in accordance with a display control data **including** an **instruction** and a parameter **sent** from other computer, a graphic pattern processing

apparatus for generating a display data in accordance...

...control means for generating an operation control signal to an operation
unit and including microprogram **memory** means for storing therein a microprogram for display control, a decoder for decoding a microinstruction
read from said microprogram **memory** and **instruction** control means **for**
controlling said decoder in accordance with the display control data; and
an operation unit including logical **address** operation means for calculating a coordinate on a display screen of a display draw point...

...pattern display control means, physical address operation means for calculating an address on said display **memory** based on the logical address and color data operation means for logically operating a selected
multi-tone information or color information and the **display** data to produce a color data....A graphic data generating apparatus comprising: information output means for outputting an image of graphic **data** ; memory
means for storing pixels of graphic data to be provided to said information
output means; a **pattern** memory for storing pattern data having at least
one bit per **pixel** of said graphic data; and a graphic data processing apparatus including a plurality of data storage means, **each** of the data
storage means storing pixel data having at least one bit, means for...

...the pixel data in said selected storage means as graphic data into an
address of **said** memory means.

...

...comprising: means for outputting graphic data of a plurality of pixels
of an image; a **display** memory connected to said outputting means for storing pixel data defining said graphic data for...

...pixels, each of said pixel data having a plurality of bits; and a graphic data **processing** apparatus performing readout of word data having
a plurality of pixel data at a word position of said **display** **memory** specified by a **source** memory address, selecting pixel data specified by a
source pixel address in the readout word data, and writing the selected pixel data in said **display** memory at a **pixel** **position** specified by a
destination pixel address of word data which **is** specified by a **destination** memory address.

...

...pattern data having at least one bit to pixel data having a

plurality of
bits; a memory **for** storing said graphic data, said **graphic data** includes at least one word, each word having a plurality of pixel data, each of...

...data having a plurality of bits; and a graphic processor having color
registers, each color **register** having stored therein pixel data having a
plurality of bits, wherein said graphic processor, responsive to said drawing instruction from said data processor, accesses **said** memory **in** units of words, selects one of said color registers based on pattern data
...

...and stores data corresponding to said pixel data stored in said selected
color register to said memory.

...

...A data processing apparatus comprising: a memory for storing graphic
data, said graphic data including at least one word, each word...

...instructions and parameters for processing said graphic data; and a graphic data processor, responsive to **an** instruction and parameters from
said data processor, for accessing **said** memory in word units, reading out
one-word graphic data designated by **a** memory address from **said memory**
, specifying at least one bit of the read-out one-word graphic data by a...

...one bit in accordance with said instruction, and writing the one-word
graphic data containing **the** processed at least one bit in **said** memory.

...

...A data processing apparatus **comprising** : **a** memory for storing graphic data, said graphic data including at least one word, each word...

...a plurality of pixels and having a plurality of bits; a graphic processor for accessing **said** memory in units of words and processing a
plurality of pixel data included in each **word** together; and a conversion
unit which includes a **plurality of** parallel-serial convertors corresponding to **the** number **of** bits within one pixel data, each parallel-serial convertor, being input bit data from each...

...said plurality of pixel data within one word according to a specified
rule, converts said **input** bit data as parallel data to serial data

and
outputs said serial data li >a memory for storing graphic data, said graphic data including at least one word, each word...

...of bits; a data processor for outputting instructions and parameters for processing graphic data, wherein **said** **instructions** include a **drawing** instruction for transferring graphic **data** stored in a predetermined location in **said** memory to another predetermined location in **said** memory; and a graphic processor, responsive to **said** drawing **instruction** and parameters corresponding to said **drawing** instruction from said data processor, for accessing **said** memory in units of words, reading out graphic data from said memory as a transfer source including a plurality of pixel data to be transferred, selecting at least one of said **pixel** data to **be** transferred **and** writing data **corresponding** to said selected pixel data into a specified location in **said** memory as **a** transfer destination according to said parameters corresponding to said drawing instructions.

...

...What is claimed is: 1. A data processing apparatus comprising: a **system** memory which holds **a** program or data; a data processor which executes said program to process said data, and generates a command or data to process graphic data; a **graphic** memory which holds a plurality of one-word graphic data, each said one-word graphic...

...pixel data arranged within a word which is a unit of data access to said **graphic** memory, and each said pixel data being constituted by plural bits; and a graphic processor which reads from said **graphic** memory graphic **data** specified **by** **a** memory address **for** specifying one-word graphic data in order to access said graphic data on a one-word basis according to a command or data from said data **processor** , specifies predetermined **pixel** data by a pixel address for specifying the predetermined pixel data in said one-word graphic **data** specified by **said** memory address, processes the pixel data thus specified according to said command, and writes one-word graphic **data** containing the pixel data thus processed in said **graphic** memory.

20/69,K/1 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0013525409 - Drawing available
WPI ACC NO: 2003-618636/200358
XRPX Acc No: N2003-492721
Digital memory for storing information in bit form and which may be accessed from two or more independent ports has locations for storing at least one information bit as several working bits
Patent Assignee: QINETIQ LTD (QINE-N)
Inventor: BURNS P D
Patent Family (2 patents, 100 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 2003067599	A1	20030814	WO 2003GB371	A	20030129	200358	B
AU 2003207002	A1	20030902	AU 2003207002	A	20030129	200425	E

Priority Applications (no., kind, date): GB 20023070 A 20020209

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 2003067599	A1	EN	24	5	

National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SC SD SE SG SK SL TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
Regional Designated States,Original: AT BE BG CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PT SD SE SI SK SL SZ TR TZ UG ZM ZW

Number	Kind	Lan	Pg	Dwg	Filing Notes
AU 2003207002	A1	EN			Based on OPI patent WO 2003067599

Alerting Abstract WO A1
NOVELTY - A digital **memory** includes separately addressable locations for storing **at least** one information **bit** as a **number** of working **bits** , and combinatorial logic output circuitry for generating the information bit from the working bits. The working bits are arranged in a number of sets, each set comprising at least one working bit for each separately addressable location.
DESCRIPTION - INDEPENDENT CLAIMS are included for:

1.a digital memory arranged as a scoreboard register with several of flag bits

2.a field programmable gate array device in which is implemented a

digital memory

3.a method of storing digital information in a memory having at least
two write ports

USE - In electronic digital storage devices using elemental logic devices
that provide a digital storage, in which the data stored within the store
may be accessed and changed from two or more independent ports.

ADVANTAGE - Allows a multiple port **memory** to be implemented using a relatively small amount of **additional** logic **functions** . Allows the **modification** of the information bit by manipulation **of** at least **one** of
the working bits, which are preferably divided into sets, where **each** set
consists of at least one working bit for each separately addressable location. Each set may be addressed for writing through a single port but
may be addressed for reading through a number of ports.

DESCRIPTION OF DRAWINGS - The drawing illustrates in block diagrammatic
form, how a single information bit may be stored as two separate working
bits.

101,102 D-type flip-flops

103 XOR gate

Title Terms/Index Terms/Additional Words: DIGITAL; **MEMORY** ; STORAGE; INFORMATION; BIT; FORM; ACCESS; TWO; MORE; INDEPENDENT; PORT; LOCATE; ONE ; WORK

Class Codes

International Classification (Main): G11C-007/10
(Additional/Secondary): G06F-013/16, G06F-013/166, G06F-013/36, G06F-013/366, G11C-008/16, G11C-008/166

File Segment: EPI;

DWPI Class: T01; U14

Manual Codes (EPI/S-X): T01-H03D; T01-H05B1; T01-H05B3; U14-A07; U14-A08B1; U14-C

Digital memory **for storing information in bit form and which may be accessed from two or more...**

Original Titles:

MULTIPLE WRITE-PORT **MEMORY**

Alerting Abstract ...NOVELTY - A digital **memory** includes separately addressable locations for storing **at least** one information **bit as** a
number of working **bits** , and combinatorial logic output circuitry **for** generating the information bit from the working bits. The...

...a digital **memory** arranged as a **scoreboard** register with several

of
flag bits a field **programmable** gate array device in which **is** implemented a digital **memory** a method of storing digital **information** in
a **memory** having at least two write ports
...

...ADVANTAGE - Allows a multiple port **memory** to be implemented **using** a
relatively small amount of **additional** logic **functions** . Allows **the** **modification** of the information bit by manipulation **of** at least **one** of
the working bits, which are preferably divided into sets, where **each** set
consists of

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
A digital **memory** for storing information **in** bit form includes a plurality of separately addressable working bits, wherein an information
bit is **stored** therein as a **combinatorial** logic **function** of **the** working **bits** . Each working bit associated with the information bit may be
addressable from a plurality of address busses. The invention provides a
multiple write-port **memory** that may be **used** in conventional fashion, or
may be conveniently adapted to implement a scoreboard register, which may
...

...to indicate the status of a logical resource. The invention is particularly suitable for implementing **memory** and scoreboard **functions**
within a **programmable** **logic** device such **as** a Field **Programmable** Gate Array. Also **disclosed** is a method of implementing a digital **memory**
.

...

...pour mettre en application les fonctions memoire et tableau d'affichage
d'un dispositif logique **programmable** comme, par exemple, un reseau de circuits prediffuses **programmables** par l'utilisateur. L'invention concerne egalement un **procede** de mise en application d'une memoire numerique.

20/69,K/2 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0012798271 - Drawing available
WPI ACC NO: 2002-654841/200270
XRPX Acc No: N2002-517374
Computer-aided semiconductor integrated circuit designing method involves defining memory **cells representing delay circuits occupying equal area and having different signal propagation delays**
Patent Assignee: ESS TECHNOLOGY INC (ESST-N)
Inventor: HERRINGTON S K; RISLER D A
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
US 6425115	B1	20020723	US 2000567862	A	20000509	200270	B

Priority Applications (no., kind, date): US 2000567862 A 20000509

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6425115	B1	EN	13	7	

Alerting Abstract US B1
NOVELTY - A pair of **memory** cells with data representing pair of delay circuits occupying equal area on a semiconductor substrate and having different propagation delay between signal input and signal output, are defined. The delay circuits have coupled p-type and n-type transistors and one of the circuit includes a capacitor coupling.
DESCRIPTION - An INDEPENDENT CLAIM is included for computer-readable medium **storing** computer-aided semiconductor **integrated** circuit design **program** .
USE - For computer-aided designing of semiconductor integrated circuits.
ADVANTAGE - Allows designers to **modify** **memory** cells to represent delay circuits having different delay time periods and occupying equal area on the semiconductor substrate.
DESCRIPTION OF DRAWINGS - The figure shows the flowchart for an iterative method of fixing timing violations using a library that allows delay cells having variable delays to be placed within a circuit to correct the violations.

Title Terms/Index Terms/Additional Words: COMPUTER; AID; SEMICONDUCTOR; INTEGRATE; CIRCUIT; DESIGN; METHOD; DEFINE; **MEMORY** ; CELL; REPRESENT; DELAY; OCCUPY; EQUAL; AREA; SIGNAL; PROPAGATE

Class Codes
International Classification (Main): G06F-017/50

US Classification, Issued: 716017000, 716002000, 716006000

File Segment: EPI;
DWPI Class: T01; U11
Manual Codes (EPI/S-X): T01-J15A2; T01-S03; U11-G01

Computer-aided semiconductor integrated circuit designing method involves defining memory cells representing delay circuits occupying equal area and having different signal propagation delays

Alerting Abstract ...NOVELTY - A pair of **memory** cells with data representing pair of delay circuits occupying equal area on a semiconductor substrate...
DESCRIPTION - An INDEPENDENT CLAIM is included for computer-readable medium **storing** computer-aided semiconductor **integrated** circuit design **program**.
...

...ADVANTAGE - Allows designers to **modify** **memory** cells to represent delay circuits having different delay time periods and occupying equal area on

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
...present invention provides a library of cells that can be stored in a computer readable **memory** and used in the computer-aided design of integrated circuits. Some of the cells in...

...n- and p-channel transistors. If so, the delay of the circuit can be further **modified** by changing the size of this capacitor. These changes in n-channel gate length and...
...to the area occupied by the unchanged circuit. Alternately, the library could allow designers to **modify** the cells such that the circuits represented by the cells differ in delay time periods...
Claims:
...than an area occupied by the first circuit;storing the first data cell in a **memory** ;generating a second data cell **comprising** second data, wherein the second data represents a second delay circuit having a second signal...

...larger than an area occupied by the second circuit;storing the second

data cell in **memory** ;wherein the first signal propagation delay is different than **the** second signal propagation delay;wherein the first area
is equal in size to the second area; andwherein the first delay circuit
comprises a first n -channel transistor coupled to a first p-channel transistor, wherein the second delay circuit comprises...

20/69,K/3 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0010776794 - Drawing available
WPI ACC NO: 2001-391513/200142
Related WPI Acc No: 2001-292756; 2001-299954; 2001-316064; 2001-397489; 2001-397490; 2001-531184; 2001-584051; 2001-591121; 2001-591122; 2002-107614; 2002-373376; 2003-842470
XRPX Acc No: N2001-288060
CAN microcontroller that supports message objects, comprises processor core
that runs CAN application
Patent Assignee: KONINK PHILIPS ELECTRONICS NV (PHIG)
Inventor: BIRNS N E
Patent Family (1 patents, 25 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 1085719	A2	20010321	EP 2000203204	A	20000915	200142	B

Priority Applications (no., kind, date): US 1999154022 P 19990915; US 2000630291 A 20000801

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 1085719	A2	EN	28	12	

Regional Designated States,Original: AL AT BE CH CY DE DK ES FI FR GB GR
IE IT LI LT LU LV MC MK NL PT RO SE SI

Alerting Abstract EP A2
NOVELTY - The CAN microcontroller comprises a processor core that runs
CAN application, message **buffers** associated with respective message objects, a CAN/CAL module that processes incoming messages including frames
and message object registers associate with each of the message objects.
USE - Supports message objects.
ADVANTAGE - **Modifies** the base address of the designated receive message
buffer by replacing the current base address with a new base address. The
microcontroller handles a message **buffer** full condition in such a manner
that ensures no loss of data. Each incoming(received) CAN Frame is automatically stored; and when writing message data into a message **buffer**
, the address will be generated automatically.
DESCRIPTION OF DRAWINGS - The drawing illustrates a high-level, **functional** block diagram of the microcontroller.

Title Terms/Index Terms/Additional Words: CAN; SUPPORT; MESSAGE; OBJECT;
COMPRISE; PROCESSOR; CORE; RUN; APPLY

Class Codes

International Classification (Main): H04L-029/06
(Additional/Secondary): G06F-015/00

File Segment: EPI;
DWPI Class: T01; W01
Manual Codes (EPI/S-X): T01-H07C1; T01-H07C5; W01-A06E1; W01-A06G2;
W01-A06X

Original Titles:
...Use of **buffer** -size mask in conjunction with address pointer to detect **buffer** -full and **buffer** -rollover conditions in a can device that employs reconfigurable message **buffers**

Alerting Abstract ...NOVELTY - The CAN microcontroller comprises a processor core that runs CAN application, message **buffers** associated with respective message objects, a CAN/CAL module that processes incoming messages including frames...
...ADVANTAGE - **Modifies** the base address of the designated receive message **buffer** by replacing the current base address with a new base address. The microcontroller handles a message **buffer** full condition in such a manner that ensures no loss of data. Each incoming(received) CAN Frame is automatically stored; and when writing message data into a message **buffer** , the address will be generated automatically...

...DESCRIPTION OF DRAWINGS - The drawing illustrates a high-level, **functional** block diagram of the microcontroller.

Original Publication Data by Authority

Original Abstracts:
...objects, and that includes a processor core that runs CAN applications, a plurality of message **buffers** associated with respective **ones** of the message objects, a CAN/CAL module that processes incoming messages that include a...

...and a plurality of message object registers associated with each of the message objects, including at least one **buffer** size register that contains a message **buffer** size value that specifies the size of the message **buffer** associated with that message **object** , and at least one **buffer** location register that contains an address pointer that points to an address of the storage location in the message **buffer** associated with that message object where the next data byte of the current incoming message is to be **stored** . The CAN/CAL module **includes** a

message handling **function** that transfers successive frames **of** the current incoming message to the message **buffer** associated with **a** selected **one** of the message objects designated as a receive message object for the current incoming message an address pointer increment **function** that, in response to **a** transfer of the current data byte to the message **buffer** associated with the designated **receive** message object, increments **the** address pointer to the address of the storage location in that message **buffer** where the next data **byte** of the current incoming message **is** to be **stored** . The CAN/CAL module **further** **includes** a frame status detection **function** that detects whether or **not** the current frame of the current incoming message is the final frame of the current incoming message, and a **buffer** -status detection **function** that, **each** time that **the** address pointer is incremented, retrieves the incremented address pointer value, retrieves the message **buffer** **size** value from the **at** **least** one **buffer** **size** **register** associated with **the** **designated** receive message object, and decodes the retrieved message **buffer** size value into a **buffer** -size mask, and determines a message **buffer** -fullness status of the **message** buffer associated with the designated **receive** message object using the retrieved incremented address pointer value and the **buffer** -size mask.

Claims:

...of message objects, comprising: a processor core that runs CAN applications; a plurality of message **buffers** associated with respective ones of **the** message objects; a CAN/CAL module that processes incoming messages that include a plurality of...

...bytes; a plurality of message object registers associated with **each** of the message objects, including: **at** **least** one **buffer** **size** register that contains a message **buffer** **size** **value** **that** **specifies** **the** **size** of the message **buffer** associated **with** **that** message object; **and** , **at** **least** one **buffer** location **register** that contains an address pointer that points to an address of **the** storage location in the message **buffer** associated with that message object where the next data **byte of** the current **incoming** message is to be **stored** ; wherein the CAN/CAL module **includes** : a message handling **function** that transfers successive **frames**

of the current incoming message to the message **buffer** associated **with** a **selected** one of the message objects designated as a receive message object for **the** current incoming message; an address pointer increment **function** that, in response to a transfer of the current data byte to the **message** **buffer** associated with the **designated** receive message object, increments the address pointer to the address of the storage location **in** **that** message **buffer** where the next data byte of the current incoming message is to be stored; a frame status detection **function** that detects whether or not the current frame of the current incoming message is the **final** frame of the current **incoming** message; and, a **buffer** -status detection **function** that: each time that the address pointer is incremented, retrieves the incremented address pointer value, retrieves the message **buffer** **size** value from **the** **at** **least** one **buffer** **size** register associated with the designated receive message object, and decodes **the** retrieved message **buffer** **size** **value** **into** a **buffer** - **size** **mask** **comprised** of **a** **plurality** x of bits, where x is equal to a prescribed number of allowable **buffer** sizes, and wherein y **bits** of the **buffer** -size mask have a first logic state and the **remaining** x-y **bits** **have** a second **logic** **state** , where 2y **equals** the retrieved message **buffer** **size** value, in terms of number of bytes; and, determines a message **buffer** -fullness status of the message **buffer** associated with the designated **receive** message object using **the** **retrieved** incremented address pointer value and the message **buffer** -size mask.

20/69,K/4 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0008719942 - Drawing available
WPI ACC NO: 1998-260871/199823
XRPX Acc No: N1998-205698

Graphic system with colour space double buffering function - writes pixel data with predetermined number of bits or with half of predetermined number of bits into pixel location or moiety of pixel location in frame buffer

Patent Assignee: 3DLABS INC LTD (THRE-N)
Inventor: HUXLEY P

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
US 5742796	A	19980421	US 1995409748	A	19950324	199823	B

Priority Applications (no., kind, date): US 1995409748 A 19950324

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5742796	A	EN	43	6	

Alerting Abstract US A

The graphic system includes one or more processors which receive **commands** from an input. Based on the received **commands** , the processors computes traffic information and writes pixel data into a frame **buffer** .
The processors are **programmed** in such a way that they write pixel data with predetermined number of bits into the pixel location of frame **buffer** or write, pixel data with half of the predetermined number of bits into a moiety of bits of the pixel location or write pixel data with half or less than the predetermined number of bits into two moieties of the pixel locations of the frame **buffer** .

A colour look up unit is provided which operates in two modes. In the first mode, predetermined number of bits from multiple location of frame **buffer** is read and display colours are generated. In the second mode, bits from the moieties are read and accordingly display colours are generated.

USE - In e.g. computer graphics and animation system of frame **buffer** e.g. for workstation, arcade games, high end simulators, and stereoscopic graphics.

ADVANTAGE - Saves **memory** space. Reduces calculation error.

Title Terms/Index Terms/Additional Words: GRAPHIC; SYSTEM; COLOUR; SPACE;
DOUBLE; BUFFER ; FUNCTION ; WRITING; PIXEL; DATA; PREDETERMINED;

NUMBER
; BIT; HALF; LOCATE; MOIETY; FRAME
Class Codes
International Classification (Main): G06F-015/16
US Classification, Issued: 395502000, 395509000, 395519000, 345189000, 345199000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-C04D; T01-J10B3B; T01-J12B

Graphic system with colour space double buffering function - ...

...**of predetermined number of bits into pixel location or moiety of pixel location in frame** buffer

Original Titles:
Graphics system with color space double **buffering** .

Alerting Abstract ...The graphic system includes one or more processors which receive **commands** from an input. Based on the received **commands** , the processors computes traffic information and writes pixel data **into** a frame **buffer** . The processors are **programmed** in such a way that **they** write pixel data with predetermined number of bits into the pixel location of frame **buffer** or write, pixel data with half of the predetermined number of bits into a moiety...

...the predetermined number of bits into two moieties of the pixel locations of the frame **buffer** .
...

...two modes. In the first mode, predetermined number of bits from multiple location of frame **buffer** is read and display colours are generated. **In** **the** second mode, bits from the moieties...

...USE - In e.g. computer graphics and animation system of frame **buffer** e.g. for workstation, arcade games, high end simulators, and stereoscopic graphics...

...ADVANTAGE - Saves **memory** space. Reduces calculation error.

Title Terms.../Index Terms/Additional Words: **BUFFER** ; ...

... **FUNCTION** ;

Original Publication Data by Authority

Original Abstracts:

A graphics subsystem which permits single **buffered** windows to exist **in** a double **buffered** system. Thus ALL **the** pixels on the screen **are** ultimately double **buffered** , but the single **buffered** should not appear **to** be double **buffered** . To support the **single** **buffered** windows, **certain** **write** **operations** are **modified** **to** **write** the **same** half-**word** of data into both the front and back half-words of an addressed location. This permits non-double **buffered** windows to remain **correct** when the RAMDAC(TM) is manipulated to swap **buffers**. >

Claims:

A graphics system, comprising:one or more processor units connected to receive **commands** from an input, **to** perform graphics computations, and to write pixel data into a frame **buffer** ;said frame **buffer** **having** a predetermined number of **data** bits per pixel;wherein said processor units are **programmable** to selectably perform operations which **include** **writing** **pixel** data with said predetermined number **of** bits into **pixel** locations of said frame **buffer** , orwriting pixel data with half or fewer of **said** predetermined number of bits into a moiety of the bits of pixel locations of said frame **buffer** , orwriting pixel data with half or fewer of said predetermined **number** of **bits** **identically** into two moieties of the bits of pixel locations of **said** frame **buffer**.>

20/69,K/5 (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0008313969 - Drawing available
WPI ACC NO: 1997-425252/199739
XRPX Acc No: N1997-354191
Non-volatile semiconductor memory **apparatus, e.g. flash EEPROM, with page latch - has latches which support programming and reading of sectors in** memory **cell array and are accessible to microcontroller**
Patent Assignee: INTEGRATED SILICON SOLUTION INC (INTE-N); NEXCOM TECHNOLOGY INC (NEXC-N)
Inventor: BAJWA A A; GANNAGE M E; WONG D K; WONG W K
Patent Family (4 patents, 72 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 1997030452	A1	19970821	WO 1997US1567	A	19970214	199739	B
AU 199719527	A	19970902	AU 199719527	A	19970214	199751	E
US 5724303	A	19980303	US 1996601963	A	19960215	199816	E
US 5862099	A	19990119	US 1996601963	A	19960215	199911	E
			US 1997939785	A	19970929		

Priority Applications (no., kind, date): US 1997939785 A 19970929; US 1996601963 A 19960215
Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 1997030452	A1	EN	15	4	

National Designated States,Original: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE HU IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK TJ TM TR TT UA UG UZ VN
Regional Designated States,Original: AT BE CH DE DK EA ES FI FR GB GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG

Number	Kind	Lan	Pg	Dwg	Filing Notes
AU 199719527	A	EN			Based on OPI patent WO 1997030452
US 5724303	A	EN	10	4	
US 5862099	A	EN			Continuation of application US 1996601963; Continuation of patent US 5724303

Alerting Abstract WO A1
The apparatus includes several page latches and a **memory** array. Each latch can be loaded and read through a data node coupled to it. The **memory** array has several cells coupled in groups to respective column lines.
The column lines are coupled to the page latches. Each of the page latches is a single stage. Alternatively, each of the page latches includes a master stage and a slave stage. Each of the master stages can be loaded and written to through a data node coupled to it. Each of the slave nodes

can be loaded from a respective master stage and read through a data node.
USE/ADVANTAGE - For use in e.g. micro-controller where use of external
SRAM is not desired. Supports **functions** normally supported by SRAM, e.g.
SRAM scratch pad. Increased speed in read- **modify** , **write** operation.

Title Terms/Index Terms/Additional Words: NON; VOLATILE; SEMICONDUCTOR;
MEMORY ; APPARATUS; FLASH; EEPROM; PAGE; LATCH; SUPPORT; **PROGRAM** ; READ;
SECTOR; CELL; ARRAY; ACCESS

Class Codes
International Classification (Main): G11C-007/00
US Classification, Issued: 365238500, 365185230, 365189050, 365230080, 365238500, 365185230, 365189050

File Segment: EPI;
DWPI Class: U13; U14
Manual Codes (EPI/S-X): U13-C04B2; U14-A03B7; U14-A07; U14-A08

Non-volatile semiconductor memory **apparatus, e.g. flash EEPROM, with page latch...**
...has latches which support programming and reading of sectors in memory
cell array and are accessible to microcontroller

Original Titles:
Non-volatile **programmable** **memory** having an SRAM capability...

...Non-volatile **programmable** **memory** having a **buffering** capability and
method of operation thereof...

Alerting Abstract ...The apparatus includes several page latches and a
memory array. Each latch can be loaded and read through a data node
coupled to it. The **memory** array has several cells coupled in groups to
respective column lines...

...use in e.g. micro-controller where use of external SRAM is not desired.
Supports **functions** normally supported by SRAM, e.g. SRAM scratch pad.
Increased speed in read- **modify** , **write** operation.

Title Terms.../Index Terms/Additional Words: **MEMORY** ; ...

... **PROGRAM** ;

Original Publication Data by Authority

Original Abstracts:
A computer system includes a computing device such as a microcontroller

and
a **memory** device. The memory **device** is illustrtively a serial device
connected to the serail port of the microcontrollerThe **memory** device includes a **page** latch load circuit which provides serial I/O to the microcontroller and transfers I/O...

...to/from the page latches. Page latches are connected over many **bit** lines
to a **memory** cell array. The **page** latches not only supports programming
and reading of sectors in the **memory** cell array, but **also** provides one
or more of the following **functions** : directly accessable to **the** microcontroller as an SRAM scratch pad, directly loadable from the **memory**
cell array to **facilitate** single byte "read- **modify** - **write** " operations,
and **loadable** **during** programming operations to support real time applications...

...A computer system includes a computing device such as a microcontroller
and a **memory** device. The **memory** device is illustratively a serial **device** connected to **the** serial port of the microcontroller. The **memory**
device includes a page latch load circuit which **provides** serial I/O to
the microcontroller and transfers I/O bits in a predetermined order to/from
the page latches. Page latches are connected over many bit lines to a **memory** cell array. The page latches not only supports **programming** and
reading of sectors in the **memory** cell array, but also provides one or more **of** the following **functions** : directly accessible to the microcontroller as an SRAM **scratch** pad, directly loadable from the **memory** cell array to facilitate single byte "read- **modify** - **write** " operations, and loadable during programming operations to **support** **real**
time applications...

...A **memory** device comprising a page latch load circuit (122) which provides serial I/O to the microcontroller (110) and transfers I/O bits
in a predetermined order to/from the page latches (124). Page latches (124)
are connected over many bit lines to a **memory** cell array (126). The page
latches (124) not only support programming and reading **of** sectors in the
memory cell array (126), but also provide one or more of the following **functions** : **directly** accessible to the microcontroller as an SRAM scratch
pad, directly loadable from **the** **memory** cell array to facilitate single
byte "read- **modify** - **write** " operations, and loadable during **programming**

operations to support real time applications.
Claims:
A non-volatile **programmable memory integrated circuit** comprising:a
plurality of data input/output ("I/O") nodes;a plurality of page latches...

...bit lines respectively coupled to the page latches;a plurality of word
lines; anda **memory** array having a plurality of non-volatile erasable-**programmable memory** cells **coupled** to respective pairs of the word lines and **bit lines**.
...

...A **memory** integrated circuit responsive to externally furnished **memory**
addresses for storing or furnishing data, comprising:a **memory array having** a plurality of non-volatile erasable- **programmable memory** cells **selectively** accessible in groups of a **common size based** on
the **memory** addresses;a high voltage circuit **coupled to** the **memory**
array; anda first **buffer having at least** a number **of storage** positions corresponding to the size of the groups, the **storage** positions
being loadable and readable externally **of** the **memory** independently of
the **memory** array and available internally to the **memory** for transferring data to the **memory** array.

20/69,K/6 (Item 6 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0008183483 - Drawing available
WPI ACC NO: 1997-286424/199726
XRPX Acc No: N1997-237141
Character generator used for generating multi-lingual text sequences - performs expansion and reduction and rotary processing on selected component data to be synthesised subsequently
Patent Assignee: SHARP KK (SHAF)
Inventor: FUJISAWA M; HASEGAWA S; IMAKI Y; IMASHIRO Y; ITO A; ITO M; KONYA M; SHIGI Y; SHIKI Y
Patent Family (8 patents, 5 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
JP 9106271	A	19970422	JP 1995265263	A	19951013	199726	B
TW 322552	A	19971211	TW 1996111785	A	19960926	199813	E
KR 1997022948	A	19970530	KR 199645587	A	19961012	199823	E
US 5771035	A	19980623	US 1996729427	A	19961011	199832	E
KR 227585	B1	19991101	KR 199645587	A	19961012	200110	E
CN 1157979	A	19970827	CN 1996122844	A	19961011	200140	E
JP 3344188	B2	20021111	JP 1995265263	A	19951013	200280	E
CN 1099096	C	20030115	CN 1996122844	A	19961011	200532	E

Priority Applications (no., kind, date): JP 1995265263 A 19951013

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
JP 9106271	A	JA	16	34	
TW 322552	A	ZH			
JP 3344188	B2	JA	15		Previously issued patent JP 09106271

Alerting Abstract JP A
The character generator has a character data **memory** (19) in which standard character data and the style of the text are stored. The component data for assembly is stored in the component data **memory** (20). The test style attribute **memory** is referred by the style recognition unit based on the text style code and character **code** input through a keyboard (11).
The angle of the stroke is obtained corresponding to the length of the processing element.
Then the text style attribute **memory** is referred to and the expansion, reduction and rotary processing are performed on selected component data.
The data synthesis module (26) forms the character data obtained by combining the selected components.
ADVANTAGE - Realizes multiple text style. Imports natural impression. Prevents interference between stroke and data components. Performs reduction processing suitable for selected character length.

Title Terms/Index Terms/Additional Words: CHARACTER; GENERATOR; GENERATE; MULTI; LINGUAL; TEXT; SEQUENCE; PERFORMANCE; EXPAND; REDUCE; ROTATING; PROCESS; SELECT; COMPONENT; DATA; SYNTHESIS; SUBSEQUENT

Class Codes
International Classification (Main): G06F-017/21, G09G-005/22, G09G-005/24, G09G-005/28
(Additional/Secondary): B41J-005/44
US Classification, Issued: 345143000, 345142000, 345471000

File Segment: EngPI; EPI;
DWPI Class: T01; P75; P85
Manual Codes (EPI/S-X): T01-J10B2; T01-J10C

Alerting Abstract ...The character generator has a character data **memory** (19) in which standard character data and the style of the text are stored. The component data for assembly is stored in the component data **memory** (20). The test style attribute **memory** is referred by the style recognition unit based on the text style **code** and character **code** input through a keyboard (11). The angle of the stroke is obtained corresponding to the...

...Then the text style attribute **memory** is referred to and the expansion, reduction and rotary processing are performed on selected component...

Original Publication Data by Authority

Original Abstracts:
...process-target element belongs, by referring to a font attribute storage based on a font **code** and character number **specified** from a keyboard. A paste component data **modifier** performs scaling up/ **down** processing and rotation processing with the selected paste component data by **referring** to the font...
Claims:
...a shape of the target portion, the character generation device comprising:font attribute storage for **storing** font attributes **including** **a** font **code** indicative of a **font** , a font name, a basic font **code** indicating a font that serves as a basis of the pertinent font, a component **code** indicating a component **to** be used for generation of the font, and **modification** information for generating **the** font;character

data storage for storing character data representing the shape of a character in the basic font;component data storage...

...character in the basic font based on the read-out character data;a component data modifier for reading out, from the component data storage,
component data to be used for the generation of the new font specified
from the input section with reference to the font attributes, and modifying the read-out component data based on the shape of the target portion so that size and disposition of the component to be used matches
the shape of the target portion; anda data synthesizer for generating
'character data of a character in the specified...

...data of the basic font read out by the shape recognizer and the component data modified by the component data modifier.>

20/69,K/7   (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0007807996 - Drawing available
WPI ACC NO: 1996-435879/199644
XRPX Acc No: N1996-367282
Packet switched cache coherent multiprocessor system - includes module
which interconnects main memory and sub-systems in accordance with interconnect control signal received from system controller
Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: COFFIN L F; EBRAHIM Z; KOHN L; LESLIE K; NISHTALA S; NORMOYLE K;
VAN L W C; VAN LOO W C
Patent Family (5 patents, 7 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 735486	A1	19961002	EP 1996302150	A	19960328	199644	B
JP 9101943	A	19970415	JP 199678714	A	19960401	199725	E
US 5634068	A	19970527	US 1995415175	A	19950331	199727	E
EP 735486	B1	20030625	EP 1996302150	A	19960328	200349	E
DE 69628778	E	20030731	DE 69628778	A	19960328	200357	E
			EP 1996302150	A	19960328		

Priority Applications (no., kind, date): EP 1996302150 A 19960328; US 1995415175 A 19950331

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 735486	A1	EN	88	18	
Regional Designated States,Original: DE FR GB IT NL SE					
JP 9101943	A	JA	87		
US 5634068	A	EN	70	18	
EP 735486	B1	EN			
Regional Designated States,Original: DE FR GB IT NL SE					
DE 69628778	E	DE			Application EP 1996302150 Based on OPI patent EP 735486

Alerting Abstract EP A1
The system has a number of sub-systems and a main **memory** coupled to a
system controller. A data-path interconnects the main **memory** and the sub-systems in accordance with interconnect control signals received from
the system controller. A number of the sub-systems have data processor which have respective chance **memory** that stores multiple blocks of data
and a set of master **cache** tags (Etags) including one tag for each data
block stored by the **cache** **memory** .
The sub-systems include a port that transmits and receives data as data
packets of a fixed **size** **equal** **in** **size** to each data block. The data
path and each port has a data path width smaller than the data block.

The
processors include a master interface, coupled to the system
controller.
ADVANTAGE - Minimises **memory** access latency to maximise
computational
throughput.

Title Terms/Index Terms/Additional Words: PACKET; SWITCH; **CACHE** ;
COHERE;
MULTIPROCESSOR; SYSTEM; MODULE; INTERCONNECT; MAIN; **MEMORY** ; SUB;
ACCORD
; CONTROL; SIGNAL; RECEIVE

Class Codes
International Classification (Main): G06F-012/08, G06F-013/00, G06F-015/163
(Additional/Secondary): G06F-015/16
US Classification, Issued: 395800000, 395468000, 395200150, 364230000,
364241800, 364260000, 364DIG001

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-H03A

Packet switched cache **coherent multiprocessor system...**

...includes module which interconnects main memory **and sub-systems in accordance with interconnect control signal received from system controller**

Original Titles:
Paketvermitteltes **cachekohaerentes** Multiprozessorsystem...

...Packet switched **cache** coherent multiprocessor system...

...Paketvermitteltes **cachekoharentes** Multiprozessorsystem...

...Packet switched **cache** coherent multiprocessor system...

...PACKET EXCHANGE-TYPE **CACHE** COHERENT MULTI-PROCESSOR SYSTEM...

...Packet switched **cache** coherent multiprocessor system.

Alerting Abstract ...The system has a number of sub-systems and a
main
memory coupled to a system controller. A data-path interconnects the
main
memory and the sub-systems in accordance with interconnect control
signals
received from the system controller. A number of the sub-systems have
data
processor which have respective chance **memory** that stores multiple
blocks
of data and a set of master **cache** tags (Etags) including one tag for
each
data block stored by the **cache** **memory** .
...

...systems include a port that transmits and receives data as data packets
of a fixed **size** equal in size to each data block. The data **path** and
each port has a data path width...

...ADVANTAGE - Minimises **memory** access latency to maximise computational
throughput.

Title Terms.../Index Terms/Additional Words: **CACHE** ; ...

... **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
A multiprocessor computer system has a multiplicity of sub-systems **and** a
main **memory** coupled to a **system** controller. An interconnect **module**,
interconnects the main **memory** and sub-systems **in** accordance **with**
interconnect control signals received from the system controller. All of
the sub-systems include a port that transmits and receives data as **data**
packets of a fixed **size** . **At** **least** two **of** **the** **sub** -systems are
data processors, each having a respective **cache** **memory** and a
respective **set** of master cache tags (Etags), including **one** **cache**
tag for each **data** block stored by the **cache** **memory** . The system
controller **maintains** a set of duplicate **cache** tags (Dtags) **for** **each**
of the data processors. The data processors each include master **cache**
logic for updating **the** master **cache** tags, while the **system** controller
includes logic for updating the duplicate **cache** tags. **Memory** transaction
request logic **simultaneously** looks up the second **cache** tag **in** each
of the sets of duplicate **cache** tags corresponding to **the** **memory**
transaction request. It **then** determines which one of the **cache**
memories and main **memory** to couple to **the** requesting data processor
based on the second **cache** states and the **address** tags stored **in the**
corresponding second **cache** tags. Duplicate **cache** **update** logic
simultaneously **updates** all of the corresponding second **cache** **tags** in
accordance **with** predefined **cache** tag update criteria.

...

...A multiprocessor computer system has a multiplicity of sub-systems and a
main **memory** coupled to a system controller. An interconnect module,
interconnects the main **memory** and sub-systems in accordance with

interconnect control signals received from the system controller. All of the sub-systems include a port that transmits and receives data as data packets of a fixed size . At least two of the sub-systems are data processors , each having a respective cache memory and a respective set of master cache tags (Etags), including one cache tag for each data block stored by the cache memory . The system controller maintains a set of duplicate cache tags (Dtags) for each of the data processors.
The data processors each include master cache logic for updating the master cache tags, while the system controller includes logic for updating the duplicate cache tags. Memory transaction request logic simultaneously looks up the second cache tag in each of the sets of duplicate cache tags corresponding to the memory transaction request.
It then determines which one of the cache memories and main memory to couple to the requesting data processor based on the second cache states and the address tags stored in the corresponding second cache tags. Duplicate cache update logic simultaneously updates all of the corresponding second cache tags in accordance with predefined cache tag update criteria. >
Claims:
...a system controller;</br> a multiplicity of sub-systems coupled to the system controller;</br> a main memory coupled to said system controller; and</br> a datapath , coupled to said system controller, interconnecting said main memory and said sub-systems in accordance with interconnect control signals received from said system controller;</br>
a plurality of said sub-systems comprising data processors, at least one of said data processors having a respective cache memory that stores multiple blocks of data and a set of master cache tags (Etags), including one Etag for each data block stored by said cache memory ;</br> at least one of said sub-systems including a port that transmits and receives data as data packets of a fixed size equal in size to said each data block; said datapath and each said port having a datapath width smaller than said each data block;</br> said at least one of said data processors including a master interface, coupled to said system controller, for sending memory transaction requests to said system controller and for receiving cache access requests from said system controller corresponding to memory transaction requests by other ones of said data processors;</br> said system controller including memory transaction request logic for processing each said memory transaction request by a requesting one

of said data **processors** , for determining which one of said **cache memories** and main **memory** to couple to the requesting data processor, for sending corresponding interconnect control signals **to** **said** datapath so **as** to couple the requesting data processor to said determined one of said **cache memories** and main **memory** , and for sending a reply message to said requesting data processor to prompt **said** **requesting** data processor to transmit/receive one data packet to/from said determined one of said **cache memories** and main **memory** .

...

...wobei mehrere der Subsysteme Datenprozessoren (102) umfassen, und wobei mindestens einer der Datenprozessoren einen entsprechenden **Cache** -Speicher (130) hat, in dem mehrere Datenblocke und ein Satz Master- **Cache** -Tags (132) gespeichert werden, die fiir jeden im **Cache** -Speicher gespeicherten Datenblock ein Master- **Cache** -Tag beinhalten,</br> wobei zumindest **eines** der Subsysteme einen Port (104) umfasst, der Daten in Form von **Datenpaketen** einer festgelegten Grosse ubertragt und empfangt, wobei die **Grosse** der Datenpakete der Grosse der **genannten** mehreren Datenblocken entspricht, und wobei der Datenweg und jeder der Ports eine Datenwegbreite haben, die...

...umfasst, uber das Speichertransaktionsanforderungen an den System-Controller verschickt und entsprechend den Speichertransaktionsanforderungen anderer Datenprozessoren **Cache** -Zugriffsanforderungen vom System-Controller empfangen werden,</br> wobei der System-Controller (110) eine Speichertransaktionsanforderungslogik beinhaltet, die...

...Speichertransaktionsanforderung eines anfordernden Datenprozessors (102) verarbeitet wird, dass sie bestimmt, welcher aus der Menge der **Cache** -Speicher (130) und des Hauptspeichers (108) mit dem anfordernden Datenprozessor verbunden werden soll, dass sie entsprechende Verbindungs-Steuersignale an den Datenweg verschickt, so dass der anfordernde Datenprozessor mit dem bestimmten **Cache** -Speicher oder dem Hauptspeicher verbunden wird, und dass sie eine Antwortnachricht an den anfordernden Datenprozessor verschickt, durch die der anfordernde Datenprozessordazu aufgefordert wird, ein Datenpaket an den bestimmten **Cache** -Speicher oder den Hauptspeicher zu ubertragen oder von **diesen** zu

empfangen,</br> wobei der System-Controller (110) fur jeden der Datenprozessoren (102) einen Cache -Tag-Zweitsatz (134) beinhaltet, wobei der Cache -Tag-Zweitsatz fur jeden Datenprozessor eine gleiche Anzahl duplizierter Cache -Tags umfasst wie der entsprechende Satz von Master- Cache -Tags (132),</br> wobei jedes Master- Cache -Tag einen Master- Cache -Tag- Cache -Zustand und ein Adress-Tag bezeichnet, und die den jeweiligen Master- Cache -Tags entsprechenden duplizierten Cache -Tags einen Zustand des duplizierten Cache -Tags und das gleiche Adress-Tag wie das entsprechende Master- Cache -Tag bezeichnen,</br> wobei die Datenprozessoren (102) jeweils eine Master - Cache -Logik fur das Aktualisieren der Master- Cache -Tags beinhalten,</br> wobei der System-Controller (110) eine Cache -Logik fur die duplizierten Cache -Tags beinhaltet, durch die die duplizierten Cache -Tags so aktualisiert werden , dass in den duplizierten Cache -Tags Zustande der duplizierten Cache -Tags und Adress-Tags gespeichert werden, die den in den Master- Cache -Tags gespeicherten Master- Cache -Tag-Zustanden und Adress - Tags entsprechen, und wobei</br> die Speichertransaktions- Anforderungslogik eine Logik fur das Prufen der duplizierten Cache -Tags umfasst, die zeitgleich die duplizierten Cache -Tags in jedem der den Speichertransaktionsanforderungen entsprechenden duplizierten Cache -Indizes liest und aufgrund der in den entsprechenden duplizierten Cache - Tags gespeicherten Zustande der duplizierten Cache -Tags und Adress-Tags bestimmt, welcher aus der Menge der Cache -Speicher (130) und des Hauptspeichers (108) mit dem anfordernden Datenprozessor verbunden werden soll, sowie eine Logik fur das Aktualisieren der duplizierten Cache -Tags, durch die zeitgleich alle entsprechenden duplizierten Cache - Tags gemass den vorab definierten Cache - Tag - Aktualisierungskriterien aktualisiert werden,</br> dadurch gekennzeichnet, dass das System so konfiguriert ist , dass</br> entweder der Master- Cache -Tag- Zustand aus dem Zustandsmuster, das im Wesentlichen aus den Zustanden Exklusiv und Modifiziert (M), Gemeinsam und Modifiziert (O), Exklusiv und Nicht Modifiziert (E), Gemeinsam und Nicht Modifiziert (S) und Ungultig (I) besteht, ausgewahlt wird, und der Zustand des duplizierten Cache - Tags aus dem Zustandsmuster, das im Wesentlichen aus Exklusiv und Modifiziert (M), Gemeinsam und Modifiziert (O), Gemeinsam und Nicht Modifiziert (S) und Ungultig (I) besteht, ausgewahlt wird , oder</br> der

Hauptspeicher (108) ein reflektierender Speicher ist, und der Master-
Cache -Tag- **Zustand** aus dem Zustandsmuster ausgewahlt wird, das im Wesentlichen aus Exklusiv und **Modifiziert** (M), Exklusiv und **Nicht** **Modifiziert** (E), Gemeinsam und Nicht **Modifiziert** (S) und **Ungultig** (I)
besteht, **und** der Zustand des duplizierten **Cache** -Tags aus dem Zustandsmuster ausgewahlt wird, das im Wesentlichen aus Exklusiv und **Modifiziert** (M), Gemeinsam und Nicht **Modifiziert** (S) und **Ungultig** (I)
besteht,</br> dass der in den duplizierten **Cache** -Tags gespeicherte Zustand des duplizierten **Cache** -Tags nie den Zustand **Exklusiv und** Nicht
Modifiziert (E) **anzeigt** , und dass,</br> wenn jeder Datenprozessor (102) Daten **modifiziert** , die in **seinem** **Cache** -Speicher (130) in einer
Cache -Line gespeichert sind, deren Master- **Cache** - **Tag** dadurch vom E-Zustand **zum** M-Zustand wechselt, der Datenprozessor keine entsprechende
Transaktionsanforderung generiert und das entsprechende **duplizierte** **Cache** -Tag unverandert in einem **Zustand** des duplizierten **Cache** -Tags verbleibt, der dem **M** -Zustand gleich ist.

A **computer** system, comprising:a system controller (110);a multiplicity of sub-systems (102) coupled to the system controller; anda
main **memory** (108) coupled to said system controller;a datapath (112), coupled to said system controller, interconnecting said main **memory** and
said sub-systems in accordance with interconnect control signals received
from said system controller...

...systems comprising data processors (102), at least one of said data processors having a respective cache **memory** (130) that stores multiple
blocks of data and a set of master cache tags (132), including one master cache tag for each data block stored by said cache **memory** ;at
least one of said sub -systems including a port (104) that transmits and
receives data as data packets of a fixed size equal in size to said
each data block, said data path and each said port having a datapath width
smaller than said each data block;said at least one of said data processors (102) including a master interface (150), coupled to said
system controller, for sending memory transaction requests to said system controller and for receiving cache access requests from said system controller corresponding to memory transaction requests by other
ones of said data processors; said system controller (110) including
memory transaction request logic arranged to process each said **memory** transaction request by a requesting one of said data processors (102), to

determine which one of said cache memories (130) and main memory (108) tocouple to the requesting data processor, to send corresponding interconnect control signals to said datapath so as to couple the requesting data processor to said determined one of said cache memories and main memory , and to send a reply message to said requesting data processor to prompt said requesting data processor to transmit/receive one data packet to/from said determined one of said cache memories and main memory ;said system controller (110) including a set of duplicate cache tags (134) for each of said data processors (102), said set of duplicate cache tags for each data processor having an equal number of duplicate cache tags as the corresponding set of master cache tags (132);each master cache tag denoting a master cache tag cache state and an address tag; the duplicate cache tag corresponding to each master cache tag denoting a duplicate cache tag state0 and the same address tag as the corresponding master cache tag;said data processors (102) each including master cache logic for updating said master cache tags;said system controller (110) including duplicate cache tag cache logic for updating said duplicate cache tags so as to store in said duplicate cache tags duplicate cache tag states and address tags corresponding to said master cache tag states and address tags stored in said master cache tags; andsaid memory transaction request logic including duplicate cache tag inspection logic for simultaneously looking up the duplicate cache tag in each of said duplicate cache indices corresponding to said each memory transaction request and for determining which one of said cache memories (130) and main memory (108) to couple to the requesting dataprocessor based on said duplicate cache tag states and said address tags stored in said corresponding duplicate cache tags, and duplicate cache tag update logic for simultaneously updating all of said corresponding duplicate cache tags in accordance with predefined cache tag update criteria, characterised in that the system is arranged such that:either said master cache tag state is selected from the set of states consisting essentially of Exclusive Modified (M), Shared Modified (O), Exclusive Clean (E), Shared Clean (S), and Invalid (I), and said duplicate cache tag state is selected from the set of states consisting essentially of Exclusive Modified (M), Shared Modified (O), Shared Clean (S), and Invalid (I) or said main memory (108) is a reflective memory , said master cache tag state is selected from the set of states consisting essentially of Exclusive Modified (M), Exclusive Clean (E), Shared Clean (S), and Invalid (I) and said

duplicate
cache **tag** state is selected from the set of states consisting essentially of Exclusive **Modified** (**M**), Shared Clean (**S**), and Invalid
(I);said duplicate cache tag state stored in said duplicate **cache** **tags**
never **indicates** said Exclusive Clean (E) state; andwhen each data processor (102) **modifies** **data** **stored** **in** **its** **cache** **memory** (130) in
a **cache** line whose master **cache** tag **thereby** transitions from **said** E
state to said M state, said data processor does not generate a corresponding transaction request and the **corresponding** duplicate **cache**
tag remains unchanged with a duplicate **cache** tag state equal to **said** M
state.

Systeme d' **ordinateur** , **comprenant** :un controleur de **systeme**
(110);une multiplicite **de** sous-systemes (102) couples au controleur de
systeme; etune memoire centrale (108) couplee au controleur de systeme;une
voie de donnees (112), couplee au **controleur** de systeme, interconnectant
la memoire centrale et les sous-systemes conformement a des signaux de
commande d'interconnexion recus du controleur de systeme;une pluralite des
sous-systemes comprenant des processeurs...

...voie de donnees et chaque port ayant une largeur de voie de donnees inferieure a **chaque** bloc de donnees;l'au moins un processeur de donnees
(102) incluant une interface principale...

...au processeur de donnees demandeur, pour envoyer vers la voie de donnees
des signaux de **commande** d'interconnexion correspondants, afin de coupler
le processeur de donnees demandeur a la memoire determinee...
...de l'ensemble correspondant d'etiquettes d'antememoire principales (132);chaque etiquette d'antememoire principale **indiquant** un etat d'antememoire d'etiquette d'antememoire principale et une etiquette d'adresse; l...

...etiquette d'antememoire principale est selectionne parmi l'ensemble d'etats consistant fondamentalement en Exclusif **Modifie** (M), Partage **Modifie** (O), Exclusif Propre (E), Partage Propre (S), et Invalide (I), et
l'etat d'etiquette d'antememoire dupliquee est selectionne parmi l'ensemble
d'etats consistant essentiellement en Exclusif **Modifie** (M), Partage **Modifie** (O), Partage Propre (S) et Invalide (I), soitla memoire principale
(108) est une memoire...

...etiquette d'antememoire principale est selectionne parmi l'ensemble d'etats consistant essentiellement en Exclusif **Modifie** (M), Exclusif Propre (E), Partage Propre (S) et Invalide (I), et l'etat d'etiquette d'antememoire dupliquee est selectionne parmi l'ensemble d'etats consistant
essentiellement en Exclusif **Modifie** (M), Partage Propre (S) et Invalide
(I);l'etat d'etiquette d'antememoire dupliquee stocke...

...n'indique jamais l'etat Exclusif Propre (E); etlorsque chaque processeur
de donnees (102) **modifie** les **donnees** stockees dans son antememoire
(130), dans une ligne d'antememoire dont l'etiquette d'antememoire principale accomplit ainsi une transition de l'etat E vers l'etat M, le processeur de donnees ne genere pas une requete de transaction correspondante, et l'etiquette d'antememoire dupliquee correspondante reste
inchangee, avec un...

...egal a l'etat M.

A computer system, comprising:a system controller;a multiplicity of sub-systems coupled to the system controller;a main **memory** coupled to said system controller; anda datapath, coupled to said **system** controller,
interconnecting said main **memory** and said sub-systems in accordance with
interconnect control signals received from said system controller...

...sub-systems comprising data processors, at least one of said data processors having a respective **cache** **memory** that stores multiple blocks
of data and a set of master **cache** tags (Etags), including one Etag for
each data block stored by said **cache** **memory** ;at least one of said sub-systems including a port that transmits and receives data as data packets of a fixed size **equal** in size to said each data block; said
datapath and each said port having a datapath width...
...of said data processors including a master interface, coupled to said
system controller, for sending **memory** transaction requests to said system
controller and for receiving **cache** access requests **from** said system controller corresponding to **memory** transaction requests by other ones of
said data processors;said system controller **including** **memory** transaction request logic for processing each said **memory** transaction request by a requesting one of said data processors, for determining which
one of said **cache** **memories** and main **memory** to couple to the requesting data **processor** , **for** sending corresponding interconnect control signals to said datapath so as to **couple** the requesting data processor to said determined one of said **cache** **memories** **and** **main** **memory** , and for sending a reply message to said requesting data

processor
to prompt said requesting data processor to transmit/receive one **data packet to** /from **said** determined one of said **cache memories and** main **memory**.>

20/69,K/8 (Item 8 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0007339785 - Drawing available
WPI ACC NO: 1995-403738/199551
Related WPI Acc No: 1998-556676
XRPX Acc No: N1995-292371
Information space image multi-view display method - involves presenting image in viewing operation area with mapping of model data items giving first and second display objects
Patent Assignee: XEROX CORP (XERO)
Inventor: DEROSE A; STONE M C
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
US 5467441	A	19951114	US 199396131	A	19930721	199551	B
			US 1994320975	A	19941006		

Priority Applications (no., kind, date): US 199396131 A 19930721; US 1994320975 A 19941006

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5467441	A	EN	76	69	Continuation of application US 199396131

Alerting Abstract US A
The method involves presenting a first image including a display object,
produced from a model structure representing a first view of the information space. The CPU presents a viewing operation area in the display
area. The CPU produces a second image representing a second view of the information space. The operation uses the present viewing position of the
viewing operation region to obtain the first model data item in the model
data structure.
The viewing operation further maps the first model data item to the first
and to a second display object not included in the first image segment. The
second image therefore includes the first and second display objects. The
CPU presents the second image in the viewing operation region so that it
overlays and replaces the first image segment. The second image is of the
same dimensions as the first and is displayed at the same time. It includes
the second display object representing information added to the information
space.
ADVANTAGE - Allows creation of spatially and temporally bounded changes
to data structure to give whit-if scenarios using original

image.Provides
clear view of complex model allowing easy data manipulation.
Title Terms/Index Terms/Additional Words: INFORMATION; SPACE; IMAGE;
DISPLAY; METHOD; PRESENT; VIEW; OPERATE; AREA; MAP; MODEL; DATA;
ITEM;
FIRST; SECOND; OBJECT

Class Codes
International Classification (Main): G06F-015/62
US Classification, Issued: 395133000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F07; T01-J10C4

Original Publication Data by Authority

Original Abstracts:
...spatial context of the first image. The method is implemented as an enhancement to the **functionality** of an application **program**, **such as a**
graphical object editor. The user requests the display of a viewing operation region (VOR) coextensively with...

...of the object-based model data structure that produced the image to produce a second **modified** view of the **portion** of the image coextensively positioned with the VOR, displaying the second, **modified** view in the **VOR**. Since the operation on the model data structure is made
to a copy of the...

...image before actually applying the changes to the model using the application. Presenting the second, **modified** image only in **the** spatial
context of the first image provides contextual feedback to the user. The
method may...
Claims:
...image definition data defining images for presentation in the display
area of the display; and **memory** for storing data; **the** data **stored**
in **the memory** including instruction data **indicating instructions** the **processor** executes and a model data structure indicating information included in the information space; the processor further being connected for accessing the data stored in the **memory**; the
method comprising: operating the processor to present a first image in a
present image position in...

...overlays and replaces the first image segment in the display area; the
second image having size and shape dimensions substantially **identical**
to **size** and shape dimensions of the viewing operation region, and

being
displayed substantially at the same time as...

20/69,K/9 **(Item 9 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

0007163381 - Drawing available
WPI ACC NO: 1995-202076/199527
XRPX Acc No: N1995-158737
Conditional memory **store from** register **pair - stores data in** memory **at**
addressable memory **locations,** several data **registers, and status** **register**
storing one status bit and arithmetic logic **unit having operand inputs** **and**
output coupled to data registers
Patent Assignee: TEXAS INSTR INC (TEXI)
Inventor: BALMER K; GUTTAG K M; KEITH B; POLAND S W
Patent Family (8 patents, 7 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 656584	A1	19950607	EP 1994308832	A	19941130	199527	B
JP 7271969	A	19951020	JP 1994296705	A	19941130	199551	E
US 5696959	A	19971209	US 1993160118	A	19931130	199804	E
			US 1995478129	A	19950607		
US 6058473	A	20000502	US 1993160118	A	19931130	200029	E
US 6173394	B1	20010109	US 1993160118	A	19931130	200104	E
			US 1999372470	A	19990811		
EP 656584	B1	20011004	EP 1994308832	A	19941130	200158	E
DE 69428499	E	20011108	DE 69428499	A	19941130	200174	E
			EP 1994308832	A	19941130		
KR 348951	B	20030124	KR 199432080	A	19941130	200339	E

Priority Applications (no., kind, date): US 1999372470 A 19990811; US 1995478129 A 19950607; EP 1994308832 A 19941130; US 1993160118 A 19931130

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
EP 656584	A1	EN	86	19	
Regional Designated States,Original: DE FR GB IT NL					
JP 7271969	A	JA	75	1	
US 5696959	A	EN	132	56	Continuation of application US 1993160118
US 6173394	B1	EN	7		Division of application US 1993160118
					Division of patent US 6058473
EP 656584	B1	EN			
Regional Designated States,Original: DE FR GB IT NL					
DE 69428499	E	DE			Application EP 1994308832
					Based on OPI patent EP 656584
KR 348951	B	KO			Previously issued patent KR 95015071

Alerting Abstract EP A1
The data processing apparatus includes memory , data registers, status

register, and arithmetic logic unit (230) coupled to data registers. An **instruction** logic circuit is connected to the addressing circuit and the
data circuit, which controls the addressing circuit and the data circuit in
response to a received **instruction** .

The logic circuit (250) controls the addressing circuit (120) and the data circuit to store data in the first register into a **specified** address
in the **memory** , if a selected status bit has a first state storing the data in a second register associated with the first, and into the **specified** address in the **memory** if the selected status bit has a second
state in response to a register pair conditional store **instruction** .

Title Terms/Index Terms/Additional Words: CONDITION; **MEMORY** ; STORAGE; REGISTER; PAIR; DATA; ADDRESS; LOCATE; STATUS; ONE; BIT; ARITHMETIC; LOGIC; UNIT; OPERAND; INPUT; OUTPUT; COUPLE

Class Codes
International Classification (Main): G06F-009/00, G06F-009/308, G06F-009/312, G06F-009/318, G06T-001/60
US Classification, Issued: 395595000, 395800000, 395566000, 712225000, 712226000, 712234000, 712226000, 712221000, 712224000, 708525000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F03A; T01-J10C5

Conditional memory store from register pair...

...**stores data in** memory **at** addressable memory **locations, several data**
registers, and status register storing one **status bit** and **arithmetic logic**
unit...

Original Titles:
...Conditional **memory** store from a register pair...

...Conditional **memory** store from a register pair...

...DEVICE FOR CONDITIONALLY STORING DATA FROM REGISTER PAIR TO **MEMORY**
...

... **Memory** store from a selected one of a register pair conditional upon
the state of a...

... **Memory** store from a register pair conditional upon a selected status
bit...

... **Instruction** having bit field designating status bits protected from
modification corresponding to arithmetic logic unit result.

Alerting Abstract ...The data processing apparatus includes **memory** , data registers, status register, and arithmetic logic unit (230) coupled to
data registers. An **instruction** logic circuit is connected to the addressing circuit and the data circuit, which controls the addressing circuit and the data circuit in response to a received **instruction** .
...

...circuit (120) and the data circuit to store data in the first register
into a **specified** address in the memory , if a selected status bit has a
first state storing the data in a second register associated with the first, and into the **specified** address in the **memory** if the selected status bit has a second state in response to a register pair conditional
store **instruction** .

Title Terms.../Index Terms/Additional Words: **MEMORY** ;

Original Publication Data by Authority

Original Abstracts:
A **memory** store operation comes from one of a pair of registers selected
by an arithmetic logic unit condition. An **instruction** logic circuit (250,
660) controls an addressing circuit (120) to store data in a first register into **memory** if a selected status bit has a first state and to
store data in a second register associated with the first register into **memory** if the selected status bit has a second state in response to a
register pair conditional store instruction . The bits may **indicate** a
negative output of the arithmetic logic unit (230), a carry out signal, an
overflow, or a zero output. The register pair conditional store **instruction** designates a particular one of the status bits to control
the conditional store. The instruction logic circuit (250, **660**) substitutes the selected status bit for a least significant bit of the register number.Thus **memory** store is from the first register if the status bit is "1" and is from the second register if the status bit is "0".
In a further embodiment the register pair conditional write **instruction**
is conditional. The **write** operation aborts if the designated condition is
true. In the preferred embodiment of this invention, the arithmetic logic
unit (230), the status register (210), the data registers (200) and the **instruction** decode logic (250, 660) are embodied in at least one digital
image/graphics processor (71) as a part of...

...A memory store operation comes from one of a pair of registers selected by an arithmetic logic unit condition. An instruction logic circuit (250, 660) controls an addressing circuit (120) to store data in a first register into memory if a selected status bit has a first state and to store data in a second register associated with the first register into memory if the selected status bit has a second state in response to a register pair conditional store instruction . The bits may indicate a negative output of the arithmetic logic unit (230), a carry out signal, an overflow, or a zero output. The register pair conditional store instruction designates a particular one of the status bits to control the conditional store. The instruction logic circuit (250, 660) substitutes the selected status bit for a least significant bit of the register number. Thus memory store is from the first register if the status bit is "1" and is from the second register if the status bit is "0". In a further embodiment the register pair conditional write instruction is conditional. The write operation aborts if the designated condition is true. In the preferred embodiment of this invention, the arithmetic logic unit (230), the status register (210), the data registers (200) and the instruction decode logic (250, 660) are embodied in at least one digital image/graphics processor (71) as a part of a multiprocessor formed in a...

...A memory store operation comes from one of a pair of registers selected by an arithmetic logic unit condition. An instruction logic circuit (250, 660) controls an addressing circuit (120) to store data in a first register into memory if a selected status bit has a first state and to store data in a second register associated with the first register into memory if the selected status bit has a second state in response to a register pair conditional store instruction . The bits may indicate a negative output of the arithmetic logic unit (230), a carry out signal, an overflow, or a zero output. The register pair conditional store instruction designates a particular one of the status bits to control the conditional store. The instruction logic circuit (250, 660) substitutes the selected status bit for a least significant bit of the register number. Thus memory store is from the first register if the status bit is "1" and is from the second register if the status bit is "0". In a further embodiment the register pair conditional write instruction is conditional. The write operation aborts if the designated

condition is true. In the preferred embodiment of this invention, the arithmetic logic unit (230), the status register (210), the data registers (200) and the instruction decode logic (250, 660) are embodied in at least one digital image/graphics processor (71) as a part of a multiprocessor formed in a single...

...of the result generated by the current arithmetic logic unit operation.

A status bit protect instruction type permits selection of status bits protected from modification corresponding to the current arithmetic logic unit result. This status bit protect instruction preferably includes individual protect bit corresponding to each status bit . If a protect bit has a first digital state, then the corresponding status bit may be modified corresponding to the current arithmetic logic unit result. If the protect bit has a second opposite digital state, then the corresponding status bit is protected from modification according to the arithmetic logic unit results.

Claims:

1. A data processing apparatus comprising: a memory storing data at addressable memory locations; an addressing circuit generating memory addresses for data accesses to said memory ; a data circuit including a plurality of data registers, a status register storing at least one status...

...having operand inputs and an output coupled to said plurality of data registers; and an instruction logic circuit connected to said addressing circuit and said data circuit, said instruction logic circuit controlling said addressing circuit and said data circuit in response to a received instruction ,said instruction logic circuit controlling said addressing circuit and said data circuit to store data in a first register into a specified address in said memory if a status bit selected from said at least one status bit has a first state and to store data in a second register associated with said first register into said specified address in said memory if a status bit selected from said at least one status bit has a second state in response to a register pair conditional store instruction .

...

...250, 245) ferner so ausgestaltet ist, daß das Statusregister (210) so

gesteuert wird, dass eine Modifikation der bestimmten dermehreren Statusbits verhindert wird, die in dem Feld des **bedingten** Registerpaar-Speicherbefehls angegeben sind...

...A data processing apparatus including a memory (20) storing data at addressable memory locations, an addressing circuit (120) generating **memory** addresses for data accesses to said **memory** , a data circuit including a plurality of data registers (200), **each** storing a predetermined number of data bits , an arithmetic logic unit (230) having operand inputs and an output coupled to said plurality...

...in accordance with the status of a prior result of said arithmetic logic unit and **instruction** logic circuit (250,245) connected to said addressing circuit and said data circuit, said instruction logic circuit **controlling** said addressing circuit and said data circuit in response to a received **instruction** ;said instruction logic circuit (250,245) being arranged to control said addressing circuit (120) and said data **registers** (200) to store said predetermined number of data bits stored in a first data register into a specified address in said **memory** if a status bit stored in said status register (210) selected from said plurality of **status** bits has a first state and to store said predetermined number of data bits stored in a second data register associated with said first data register into said specified address in said **memory** if said selected status bit stored in said status register (210) has a second state **in** response to a register pair conditional store **instruction** ,wherein **said** register pair conditional store instruction **includes** a field of a plurality of bits ("N C V Z" Figure 19) designating whether particular ones of said plurality of status bits are protected from being set corresponding to said result of said arithmetic logic unit; andsaid **instruction** logic circuit (250,245) is further arranged to control said status register (210) to prevent modification of said particular ones of said plurality of status bits designated in said field of said register pair conditional store instruction .

...

...Dispositif de traitement de donnees incluant une memoire (20) **memorisant** des donnees en des emplacements de memoire adressables, un circuit d'adressage (120) produisant des adresses de memoire pour des

acces de donnees a ladite memoire, un circuit de donnees comprenant une pluralite de registres de donnees (200), dont chacun memorise un nombre predetermine de bits de donnees, une unite arithmetique et logique (230) comportant des entrees d'operandes et une sortie couplee a ladite pluralite de registres de donnees, un registre d'etat (210) memorisant une pluralite de bits d'etat positionnes en fonction de l'etat d'un resultat anterieur de ladite unite arithmetique et logique et un circuit logique d'instructions (250, 245) connecte audit circuit d'adressage et audit circuit de donnees, ledit circuit logique d'instructions commandant ledit circuit d'adressage et ledit circuit de donnees en reponse a une instruction recue;ledit circuit logique d'instructions (250, 245) etant agence de maniere a commander ledit circuit d'adressage (120) et lesdits registres de donnees (200) pour memoriser ledit nombre predetermine de bits de donnees memorises dans un premier registre de donnees, a une adresse specifiee dans ladite memoire si un bit d'etat memorise dans ledit registre d' etat (210) selectionne a partir de ladite pluralite de bits d'etats possede un premier etat, et pour memoriser ledit nombre predetermine de bits de donnees memorises dans un second registre de donnees associe audit premier registre de donnees, a ladite adresse specifiee dans ladite memoire si ledit bit d' etat selectionne memorise dans ledit registre d'etat (210) possede un second etat en reponse a une instruction de memorisation conditionnelle dans une paire de registres,dans lequel ladite instruction de memorisation conditionnelle dans une paire de registres comprenant une pluralite de bits ("N C V Z", figure 18) indiquant si des bits particuliers de ladite pluralite de bits d'etat sont proteges vis -a-vis d'un positionnement correspondant audit resultat de ladite unite arithmetique et logique; etledit circuit logique d'instructions (250, 245) est en outre agence de maniere a commander ledit registre d'etat (210) pour empecher une modification desdits bits particuliers faisant partie de ladite pluralite de bits d'etat designe dans ladite zone de ladite instruction de memorisation conditionnelle dans une paire de registres.

A data processing apparatus comprising: a memory storing data at addressable memory locations;an addressing circuit generating memory addresses for data accesses to said memory; a data circuit includinga

data register file including a plurality of data registers, each of said plurality of data registers storing a predetermined number of data bits,an arithmetic logic unit having operand inputs and an output coupled to said plurality of data registers, said arithmetic logic unit generating at least one status bit corresponding to said output, anda status register connected to said arithmetic logic unit for storing said at least one type of status bit; andan instruction logic circuit connected to said addressing circuit and said data circuit, said instruction logic circuit controlling said addressing circuit and said data circuit in response to a received instruction , said instruction logic circuit controlling said addressing circuit and said data circuit to store said predetermined number of data bits stored in a first data register into a specified address in said memory if a status bit selected from said at least one type of status bit has a first state and to store said predetermined number of data bits stored in a second data register associated with said first data register into said specified address in said memory if a status bit selected from said at least one type of status bit has a second state in response to a register pair conditional store instruction .

A data processing apparatus comprising: a memory storing data at addressable memory locations;an addressing circuit generating memory addresses for data accesses to said memory ;a data circuit includinga plurality of data registers, each storing a predetermined number of...

...an arithmetic logic unit having operand inputs and an output coupled to said plurality of data registers, said status register sets status bits corresponding to said output of said arithmetic logic unit;an instruction logic circuit connected to said addressing circuit and said data circuit, said instruction logic circuit controlling said addressing circuit and said data circuit in response to a received instruction , said instruction logic circuit controlling said addressing circuit and said data circuit to store said predetermined number of data bits stored in a first data register into [illegible] specified address in said memory it a status bit selected from said plurality of different types of status bit has a first state and to store said predetermined number of data bits stored in a second data register associated with said first data register into said specified address in said memory if a status bit selected from said plurality of different types of status bit has a second state in response to a register pair conditional store instruction ; andsaid register pair conditional store instruction

including a plurality of bits designating whether **particular ones of said**
plurality of different types of

20/69,K/10 (Item 10 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0007089731 - Drawing available
WPI ACC NO: 1995-115926/199516
Related WPI Acc No: 1995-053872
XRPX Acc No: N1995-091476
Object based model data structure operating appts for producing second image related to first image - uses viewing operation region to select portion of original image for which second image is wanted
Patent Assignee: XEROX CORP (XERO)
Inventor: DEROSE A; STONE M C
Patent Family (3 patents, 2 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
CA 2124604	A	19950122	CA 2124604	A	19940530	199516	B
US 5596690	A	19970121	US 1993096200	A	19930721	199710	E
CA 2124604	C	19990413	CA 2124604	A	19940530	199933	E

Priority Applications (no., kind, date): US 199396200 A 19930721

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
CA 2124604	A	EN	189	63	
US 5596690	A	EN	78	61	
CA 2124604	C	EN			

Alerting Abstract CA A
The appts includes an output circuitry connected to a display having a display area presenting images. The display area has a first image displayed in a present image position in it. The first image includes a first display object having a present object position in the first image. A processor is connected for receiving the signals from the signal source.
The processor also provides image definition for defining images to the output circuitry. A memory is also included for storing data. E.g. the data stored in the memory includes instruction data indicating instructions the processor executes and a first image data structure used for producing the first object data item represented by the first display object in the first image.
USE/ADVANTAGE - For operating processor controlling machine fitted with display for displaying images either static or animated. Capable for access or manipulating data and information that is not currently represented by display , currently visible in original image.

Title Terms/Index Terms/Additional Words: OBJECT; BASED; MODEL; DATA; STRUCTURE; OPERATE; APPARATUS; PRODUCE; SECOND; IMAGE; RELATED; FIRST;

VIEW; REGION; SELECT; PORTION; ORIGINAL

Class Codes
International Classification (Main): G06F-003/14, G06T-015/00
(Additional/Secondary): G06F-015/70, G06T-001/00
US Classification, Issued: 395133000, 395135000, 395120000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F07; T01-J12B

Alerting Abstract ...The processor also provides image definition for defining images to the output circuitry. A **memory** is also included for storing data. E.g. the data stored in the **memory** **includes** **instruction** data indicating instructions the processor executes and a first image data structure used for producing the first object...

Original Publication Data by Authority

Original Abstracts:
...object-based model data structure that produced the graphical object image to produce a second modified view of the portion of the graphical object image coextensively positioned with the VOR, displaying the second **modified** view in the VOR. >

Claims:
...source, and connected for providing image definition data defining images to the output circuitry; and memory for storing data; the data **stored** in the memory including instruction data **indicating** **instructions** the processor executes; and a **first** image model data structure used by a model-based operation to produce the first image; the first image model data structure...

...the first image;the processor further being connected for accessing the data stored in the memory ;the method comprising:operating the processor to receive request signal data from the signal source **indicating** a display request to present a viewing operation region in a present viewing position in...
...the first object data item represented by the first display object; the second image having size and shape dimensions substantially identical to **size** and shape dimensions of the viewing operation region and including a second display object showing a **modified** view **of** the first display object; and providing the image definition data defining the second image to the output circuitry connected to the display so **that** the

display presents the second image in the viewing operation region substantially at the same...

...of the second display object is outside the boundary of the viewing operation region, the modified view of the first display object is clipped to the boundary of the viewing operation region and only the first
portion of the second display object is shown in the second image as the
modified view of the first display object;presentation in the viewing operation region of the second image produced using the first object data
item giving a perception to the machine user of presenting in the second
image a modified view of the first image segment in the spatial context
of the first image.

20/69,K/11 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPTX

0007089695
WPI ACC NO: 1995-115885/199516
XRPX Acc No: N1995-091441
Absolute static lock for files and directories on magnetic disk storage media - reading directory entry data field on disk for target file into memory , restructuring directory entry data field in non-DOS format, and replacing original directory entry data field on target media
Patent Assignee: YEOW K (YEOW-I)
Inventor: YEOW K
Patent Family (3 patents, 2 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
CA 2101123	A	19950123	CA 2101123	A	19930722	199516	B
US 5557674	A	19960917	US 1994342169	A	19941118	199643	
NCE							
CA 2101123	C	19971230	CA 2101123	A	19930722	199812	E

Priority Applications (no., kind, date): US 1994342169 A 19941118; CA 2101123 A 19930722

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
CA 2101123	A	EN	29	2	
US 5557674	A	EN	10	2	
CA 2101123	C	EN			

Alerting Abstract CA A
To apply absolute static lock at media level on a target file or directory, the directory entry data field on disk for the target file or directory on the host machine is located and read into a convenient area of the host machine memory . The directory entry data field is restructured and in non-DOS format. The original directory entry data field on the target media is replaced with the restructured non-DOS directory entry data field.

Encryption of the target file contents may be incorporated into the absolute lock process if required. Target files or directories upon which the absolute static lock has been successfully applied cannot be accessed by DOS at media level, for the critical operations of read, copy, overwrite and erase. In the reverse unlock process, the previously applied absolute static lock is removed from a target file or directory restoring it to the original unlocked DOS state. If the target media is a floppy disk, absolute static lock to the floppy disk can be applied or removed.

USE/ADVANTAGE - Absolute static lock may be applied at media level, to
files and directories in FAT-based storage media, of single machine personal microcomputers running within Disk Operating System (DOS) or equivalent environment. Cannot be read, copied, over-written or erased. Is
transparent to DOS. Is achieved without controlling file, and without occupying additional sector space on target disk.

Title Terms/Index Terms/Additional Words: ABSOLUTE; STATIC; LOCK; FILE; DIRECTORY; MAGNETIC; DISC; STORAGE; MEDIUM; READ; ENTER; DATA; FIELD; TARGET; MEMORY ; RESTRUCTURING; NON; FORMAT; REPLACE; ORIGINAL

Class Codes
International Classification (Main): G06F-012/14, G06F-009/445
US Classification, Issued: 380004000, 380025000, 364DIG001, 364246600, 364246800, 364246900, 364DIG002, 364969000, 364969400, 364969300

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-H01C2; T01-J12C

...**reading** directory entry data field on disk **for target file into** memory
, **restructuring** directory entry data field in non-DOS **format, and replacing**
original directory entry data...

Alerting Abstract ...the host machine is located and read into a convenient area of the host machine memory . The directory entry data field is restructured and in non-DOS format. The original directory...

Title Terms.../Index Terms/Additional Words: MEMORY ;

Original Publication Data by Authority

Original Abstracts:
...the host machine is located and read into a convenient area of the host
machine memory . The directory entry data field is restructured according to the procedure and in the non-DOS format of this...
Claims:
...compatible computer, for operations including read, copy, overwrite and
erase, comprising the steps of: (a) modifying directory entry field for
the target file into a special null format, including alteration of the
directory entry data for filesize, and file...

...specifically as a null dir file possessing null filesize and null starting cluster bytes; (b) said modification including storage of data in
encrypted form at predetermined byte offset positions within the directory entry field, said data being completely transparent to

the operating system, and said offset or equivalent positions not normally used by, or are functionally transparent to, the operating system of the computer; (c) said stored data including data on summary bytes for the access password accompanying the user request to lock the...

...the target disk media is a floppy disk, suitably adjusting the track layout parameters permitting program means of said method in conjunction with BIOS means of said computer to reformat the entire outermost track of the floppy disk into a predetermined, non-standard...

...the disk occupying first sector position in said track; (g) said track format also including at least a non-standard sector **size** for the boot sector.

20/69,K/12 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0006482670 - Drawing available
WPI ACC NO: 1993-288609/199336
XRPX Acc No: N1993-221950
Digital data processor instruction pre-fetch unit - has branch history table indicates occurrence of branch instruction having target address that was previously taken
Patent Assignee: SAMSUNG ELECTRONICS CO LTD (SMSU); WANG LAB INC (WANG)
Inventor: SABA J A; SCHWARTZ M J; TANG-KONG R; TANK-KONG R
Patent Family (9 patents, [illegible] countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 1993017384	A1	1993[illegible]2	WO 1992US6813	A	19920813	199336	B
AU 199224723	A	1993[illegible]	AU 199224723	A	19920813	199403	E
EP 628184	A1	[illegible]941214	EP 1992918413	A	19920813	199503	E
			WO 1992US6813	A	19920813		
US 5404467	A	19950404	US 1992843868	A	19920227	199519	E
			US 1994261318	A	19940616		
JP 7504520	W	19950518	WO 1992US6813	A	19920813	199528	E
			JP 1993514792	A	19920813		
AU 665368	B	1996[illegible]04	AU [illegible]24723	A	19920813	199608	E
EP 628184	B1	[illegible]	EP [illegible]2918413	A	19920813	199847	E
			WO 1992US6813	A	19920813		
DE 69227465	E	[illegible]	DE [illegible]465	A	19920813	199903	E
			EP [illegible]	A	19920813		
			WO 1992US6813	A	19920813		
JP 3423310	B2	2003[illegible]07	WO 1992US6813	A	19920813	200345	E
			JP 1993514792	A	19920813		

Priority Applications (no., kind, date): US 1994261318 A 19940616; US 1992843868 A 19920227

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 1993017384	A1	EN	[illegible]		

National Designated States, Original: AU CA JP
Regional Designated States, Original: AT BE CH DE DK ES FR GB GR IE IT LU MC NL SE

Number	Kind	Lan	Pg	Dwg	Filing Notes
AU 199224723	A	EN			Based on OPI patent WO 1993017384
EP 628184	A1	EN	[illegible]	1	PCT Application WO 1992US6813; Based on OPI patent WO 1993017384

Regional Designated States, Original: BE DE FR GB NL

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5404467	A	EN	[illegible]	5	Continuation of application US 1992843868
JP 7504520	W	JA	18		PCT Application WO 1992US6813; Based on OPI patent WO 1993017384
AU 665368	B	EN			Previously issued patent AU 9224723; Based on OPI patent WO 1993017384

EP 628184 B1 EN PCT Application WO 1992US6813
Based on OPI patent WO 1993017384
Regional Designated States,Original: BE DE FR GB NL
DE 69227465 E DE Application EP 1992918413
PCT Application WO 1992US6813
Based on OPI patent EP 628184
Based on OPI patent WO 1993017384
JP 3423310 B2 JA 26 PCT Application WO 1992US6813
Previously issued patent JP
07504520

Based on OPI patent WO 1993017384

Alerting Abstract WO A1

The **instruction** pre-fetch unit includes a sequential **instruction** physical address generator, an instruction cache and an **instruction** queue having multiple registers. Some of the instructions are Branch **instructions** having an associated Target Address.

A Branch History Table (BHT) has an input responsive to the **instruction** physical address. The BHT output indicates whether, during a preceding execution of a corresp. Branch instruction output by the **instruction** **cache**, the execution of the corresp. Branch instruction resulted in subsequent program execution being redirected to the Target Address associated with the corresp. Branch instruction.

USE/ADVANTAGE - For high performance CPU with pipe-lined **instruction** execution and virtual addressing capabilities. Optimises efficiency of **instruction** unit.

Equivalent Alerting Abstract US A

Multiple Branch mark bits are stored in an instruction **queue**, on a half word basis, in conjunction with a double word of **instruction** data that is prefetched from an instruction cache. The Branch Target Address is employed to redirect instruction prefetching. The Branch Target Address is also pipelined and follows the associated Branch **instruction** through an instruction pipeline. The prefetch unit includes circuitry for automatically self-filling the instruction pipeline.

During a Fetch stage a previously generated Virtual Effective Address is applied to a translation buffer to generate a physical address which is used to access a data cache. The translation buffer includes two translation buffers, with the first translation buffer being a reduced subset of the second. The first translation buffer is probed, during a Generate stage, to prefetch, if possible, a required operand. The prefetch unit further provides 24-bit or 31-bit effective address generation on an instruction by instruction basis.

USE - Prefetch unit which includes a branch history table for providing

an indication of an occurrence of a Branch instruction having a Target Address that was previously taken.

Title Terms/Index Terms/Additional Words: DIGITAL; DATA; PROCESSOR; INSTRUCTION ; PRE; FETCH; UNIT; BRANCH; HISTORY; TABLE; INDICATE; OCCUR; TARGET; ADDRESS

Class Codes
International Classification (Main): G06F-009/38
(Additional/Secondary): G06F-009/26, G06F-009/32
US Classification, Issued: 395375000, 364DIG001, 364243420, 364231800, 364247300, 364DIG002, 364938000, 364955600, 364964260, 364261700, 364263100

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F03A; T01-H03A
Digital data processor instruction pre-fetch unit...

...**has branch history** table indicates occurrence of **branch instruction having target address** that was previously taken

Original Titles:
...CPU HAVING PIPELINED INSTRUCTION UNIT AND EFFECTIVE ADDRESS CALCULATION UNIT WITH RETAINED VIRTUAL ADDRESS CAPABILITY...

...UNITE CENTRALE AYANT UNE UNITE D' INSTRUCTIONS A TRAITEMENT PIPELINE ET
UNE UNITE DE CALCUL D'ADRESSES EFFECTIVE A CAPACITE D'ADRESSES...

...CPU HAVING PIPELINED INSTRUCTION UNIT AND EFFECTIVE ADDRESS CALCULATION UNIT WITH RETAINED VIRTUAL ADDRESS CAPABILITY...

...UNITE CENTRALE AYANT UNE UNITE D' INSTRUCTIONS A TRAITEMENT PIPELINE ET
UNE UNITE DE CALCUL D'ADRESSES EFFECTIVE A CAPACITE D'ADRESSES...

...CPU having pipelined instruction unit and effective address calculation unit with retained virtual address capability...

...CPU HAVING PIPELINED INSTRUCTION UNIT AND EFFECTIVE ADDRESS CALCULATION UNIT WITH RETAINED VIRTUAL ADDRESS CAPABILITY

Alerting Abstract ...The instruction pre-fetch unit includes a sequential **instruction** physical address [illegible], an **instruction cache** and an instruction queue [illegible] registers. Some of the **instructions** are Branch instructions having an associated Target Address
...

...A Branch History Table (BHT) has an input responsive to the **instruction** physical address. The BHT output indicates whether, during a preceding execution of a corresponding Branch instruction, [illegible] by the **instruction**

cache , the execution of the corresp. Branch **instruction** resulted in subsequent program execution being redirected to the Target Address associated with the corresp. Branch instruction .

...

...USE/ADVANTAGE - For high performance CPU with pipe-lined **instruction** execution and virtual addressing capabilities. Optimises efficiency of **instruction** unit.

Equivalent Alerting Abstract ...Multiple branch mark bits are stored in an **instruction** queue , on a half word basis, in conjunction with a double word of instruction data that is prefetched from an **instruction** **cache** . The Branch Target Address is employed to redirect **instruction** prefetching. The Branch Target Address is also pipelined and follows the associated Branch instruction through an instruction pipeline. The prefetch unit includes circuitry for automatically self-filling the **instruction** pipeline...

...During a Fetch stage a previously generated Virtual Effective Address is applied to a translation buffer to generate a physical address which is used to access a data cache . The translation buffer includes two translation buffers , with the first translation buffer being a reduced subset of the second. The first translation buffer is probed, during a Generate stage, to prefetch, if possible, the required operand. The prefetch unit further provides 24 bit or 31-bit effective address generation on an instruction by instruction basis...

...includes a Branch history table for providing an indication of an occurrence of a Branch instruction having a Target Address that was previously taken.

Title Terms.../Index Terms/Additional Words: INSTRUCTION ;
Original Publication Data by Authority

Original Abstracts:
...includes a Branch history table for providing an indication of an occurrence of a Branch instruction having a Target Address that was previously taken. A plurality of Branch mark bits are stored in an **instruction** queue , on a half word basis, in conjunction with a double word of instruction data that is prefetched from an **instruction** **cache** . The Branch Target Address is employed to redirect **instruction** prefetching. The Branch Target Address is also pipelined and follows the associated Branch instruction through an **instruction** **pipeline** . The prefetch unit includes circuitry for automatically self-filling the instruction pipeline. During a Fetch stage a previously generated Virtual Effective Address is applied to a

translation buffer to generate a physical address which is used to access a data cache . The translation buffer includes a first and a second translation buffer ,with the first translation buffer being a reduced subset of the second. The first translation buffer is probed, during a Generate stage, to prefetch, if possible, the required operand. The prefetch unit further provides 24-bit or 32-bit effective address generation on an instruction by instruction basis.

...

...includes a Branch history table for providing an indication of an occurrence of a Branch instruction having a Target Address that was previously taken. A plurality of Branch mark bits are stored in an instruction queue , on a half word basis, in conjunction with a double word of instruction data that is prefetched from an instruction cache . The Branch Target Address is employed to redirect instruction prefetching. The Branch Target Address is also pipelined and follows the associated Branch instruction through an instruction pipeline. The prefetch unit includes circuitry for automatically self -filling the instruction pipeline. During a Fetch stage a previously generated Virtual Effective Address is applied to a translation buffer to generate a physical address which is used to access a data cache . The translation buffer includes a first and a second translation buffer , with the first translation buffer being a reduced subset of the second. The first translation buffer is probed, during a Generate stage, to prefetch, if possible, the required operand. The prefetch unit further provides 24-bit or 32-bit effective address generation on an instruction by instruction basis.

...

...includes a Branch history table for providing an indication of an occurrence of a Branch instruction having a Target Address that was previously taken. A plurality of Branch mark bits are stored in an instruction queue , on a half word basis, in conjunction with a double word of instruction data that is prefetched from an instruction cache . The Branch Target Address is employed to redirect instruction prefetching. The Branch Target Address is also pipelined and follows the associated Branch instruction through an instruction pipeline. The prefetch unit includes circuitry for automatically self-filling the instruction pipeline. During a Fetch stage a previously generated Virtual Effective Address is applied to a translation buffer to generate a physical address which is used to access a data cache . The

translation buffer includes a first and a second translation buffer ,with the first translation buffer being a reduced subset of the second. The first translation buffer is [illegible], during a Generate stage, to prefetch, if possible, the required operand. The prefetch unit further provides [illegible] bit or [illegible]-bit effective address generation on an
instruction by instruction basis.
Claims:
The instruction pre-fetch unit includes a sequential instruction physical address generator, an instruction cache and an instruction
queue having multiple registers. Some of the instructions are Branch instructions having an associated Target Address...

...A Branch History Table (BHT) has an input responsive to the instruction
physical address. The BHT output indicates whether, during a preceding execution of a corresponding Branch instruction [illegible] by the instruction cache , the execution of the corresponding Branch instruction resulted in subsequent program execution being redirected to the Target Address associated with the corresponding Branch instruction.

...

...Befehlscachespeicher ausgegebenen entsprechenden Sprungbefehls die genannte Ausfuehrung des [illegible] Sprungbefehls zur Umleitung einer nachfolgenden Programmausfuehrung zu einer Zieladresse
gefuehrt hat, die dem [illegible] entsprechenden Sprungbefehl zugeordnet war
, gekennzeichnet durch [illegible] - eine Befehls-[illegible]-line-Einrichtung (92, 94,
96, 98), die von einer [illegible] ...

...use in a digital data processor (10) for prefetching individual ones of
a plurality of instructions prior to execution of the instructions , said digital data processor including /br> means (40) for generating sequential ones of a plurality of instruction addresses,</br>
instruction cache means (78) having an input responsive to said instruction addresses and an output for providing a corresponding instruction ,</br> instruction queue means (80, 82, 84, 86) having an
input coupled to said output of said instruction cache means and having a plurality of registers [illegible] a width sufficient
to store at least [illegible] , instruction execution means (16), [illegible]</br> [illegible] history table means (42) having an input responsive to said instruction physical addresses and
an output for providing an indication [illegible], during a preceding
execution of a [illegible] output by said instruction cache [illegible] corresponding Branch instruction resulted in subsequent program execution being redirected to
a Target Address [illegible] with said corresponding Branch

instruction ,
characterized by</br> [illegible] instruction pipeline means (92, 94, 96, 98)
comprised of a plurality of serially coupled register **means** (92 ,
94,
96) each having a width sufficient to store a widest **possible**
instruction , said instruction pipeline [illegible] having an input coupled
to
an output of said [illegible] queue [illegible] (80 , 82, 84, 86) and
an
output for coupling [illegible] instruction [illegible] means (36);</br>
wherein some of said instructions are [illegible] instructions having an
associated Target Address, and</br> wherein each of said register
means of said instruction queue means [illegible], 84, **86**) **and each**
of
said serially coupled register means (92, 94, 96) of said **instruction**
pipeline means ([illegible]) further include means **for** **storing**
a
Mark indication output from said Branch history table means (42) and
for
advancing said stored Mark indication through said **instruction**
queue means and through said instruction [illegible] line means in
association
with a corresponding [illegible] instruction [illegible] means (44) **for**
generating effective [illegible] wherein
certain
of said **instructions** [illegible] a specific [illegible] or ones of a **plurality**
of
general register means [illegible] wherein the effective address
generating
means (44) is comprised [illegible] 60) for **storing** a
duplicate copy of said plurality of general register means, said
storing
means (60) being responsive to prefetched instructions (98, 100) for
outputting a specific [illegible] ones of said general
register means,</br> [illegible] calculating an effective
address of an operand [illegible] **instruction** , in
cooperation with the [illegible] copies of said
general
register means output [illegible] said storing [illegible]),</br> **wherein**
said storing means [illegible] outputs of said
execution
means (32) for...

...copy of said general register [illegible] storage of
information in said general register means [illegible] bypassing means
(62,
64, 66), responsive [illegible] previously prefetched
instruction that [illegible] said previously
prefetched instruction [illegible] means (60), **for**
selectively bypass [illegible]

...means (36) for [illegible] address of an operand
associated with an [illegible] subsequent to said
instruction that [illegible] means; and .</br>
suspending means [illegible] of a **previously**
prefetched instruction [illegible] execution of said
previously prefetched [illegible] general register

means, for suspending an occurrence of said effective address calculation for an instruction contained in one of said register means (92) until an execution of said previously prefetched instruction modifies said content of said general register means.

Apparatus for use in a digital data processor for prefetching individual ones of a plurality of instructions prior to execution of said instructions, said apparatus comprising: means for generating sequential ones of a plurality of instruction physical addresses; instruction cache means having an input responsive to said instruction physical addresses and an output for providing a corresponding instruction; means for generating an effective address of an operand or operands associated with certain ones of said instructions that are output from said instruction cache means; instruction queue means having an input coupled to said output of said instruction cache means and having a plurality of register means, each of said register means having a width sufficient to store one instruction or a portion of one instruction; instruction pipeline means comprised of a plurality of serially coupled instruction register means each having a width sufficient to store a largest possible instruction, said instruction pipeline means having an input coupled to an output of said instruction queue means and an output for coupling to instruction execution means, said plurality of serially coupled instruction register means including, first instruction register means (IRG), having an input coupled to said output of said instruction queue means, for storing an instruction during an effective address generation operation for said instruction; second instruction register means (IRF), having an input coupled to an output of said IRG register means, for storing said instruction during an operand fetch operation for said instruction; and third instruction register means (IRE), having an input coupled to an output of said IRF register means and an output coupled to said instruction execution means, for storing said instruction during said instruction execution operation; wherein some of said instructions are Branch instructions having an associated Target Address, and wherein said apparatus further includes, Branch history table means having an input responsive to said instruction physical addresses and an output for providing an indication of whether,

during a preceding execution of a [illegible]sponding Branch instruction output by said instruction cache [illegible] said [illegible] corresponding Branch instruction res[illegible] sequent[illegible] [illegible]ition being redirected to said Target Address associated with [illegible] [illegible]sponding Branch instruction , said Branch history table means being comprised of Target Address storage means for [illegible] outputting a pre[illegible] Target Address associated with a [illegible]ding branch in[illegible] output by said instruction cache means; wherein each of said register means of said instruction queue means and said instruction pipeline means includes means for storing said indication output by said Branch history table means [illegible] advancing said [illegible] indication through said instruction [illegible] means and [illegible] instruction pipeline means in association with [illegible] corresponding [illegible] instruction ; wherein said effective address [illegible] [illegible]lurality of serially coupled address register means comprising, first address register means (IAG) for storing a virtual instruction address during an effective address generation operation for an instruction stored within said IRG register means; second address register means (IAF), having an input coupled to an output of said IAG register means , for storing said virtual instruction address during said operand prefetch operation for an instruction stored [illegible] IRF register means; and third address [illegible] input coupled to an output of said [illegible] virtual instruction address during said [illegible] of an instruction stored within said IRE register means, wherein [illegible] effective [illegible] means further includes, fourth address register means [illegible] an input coupled to an output of said IAE register means, for [illegible] said virtual instruction address [illegible] instruction is a Branch instruction that fails [illegible] execution of said Branch instructio[illegible] [illegible]ising, first target address register [illegible] having [illegible] coupled to an output of said Branch history [illegible] means, for storing a Target Address of [illegible] instruction [illegible] within said instruction queu[illegible] , said TA[illegible] means having an output coupled to said [illegible] for providing said Target Address th[illegible] register means (TARIRG), having an input coupled [illegible] register means,

for
...

...Address during an effective address formation operation for said
Target
Address associated with a branch instruction [illegible] is stored within
said
IRG register means ; third target address register means (TARIRF),
having an input coupled to an output of said [illegible]...

...means, for storing said Target Address during an operand prefetch
operation associated with a branch instruction that is stored within
said
IRF register means; and fourth target address register means (TARIFE),
having an input coupled to an output of said [illegible] register means ,
for
storing said Target Address associated with a Branch instruction that
is
stored within said IRE register means during an execution of said
Branch
instruction.>

20/69,K/13 (Item 13 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0006396827 - Drawing available
WPI ACC NO: 1993-197360/199324
XRPX Acc No: N1993-1517[illegible]
Document image compression method establishing required **memory space -**
selects quantising matrix in dependence of [illegible] space **needed to store**
image data after compression by discrete transform technique.
Patent Assignee: UNISYS CORP (BURS)
Inventor: HIGGINS-LUTHMAN [illegible] J; HIGGINS-LUTHMAN [illegible] J; KIDD R C; **KLEIN R D;**
YEN R; YEN R C; [illegible] M
Patent Family (11 [illegible] 13 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 1993011629	[illegible]	[illegible]0610	WO 1992US9910		19921119	199324	B
EP 568681	[illegible]	[illegible]1110	EP 1992925384	[illegible]	19921119	199345	E
			WO 1992US9910	A	19921119		
US 5339368	[illegible]	[illegible]	US [illegible]	[illegible]	19911121	199432	E
JP 7502154	W	[illegible]	WO [illegible]10		19921119	199517	E
			JP [illegible]510145	[illegible]	19[illegible]119		
EP 798919	A2	[illegible]	EP [illegible]	[illegible]	[illegible]119	199744	E
			EP [illegible]1057[illegible]	[illegible]	19[illegible]1119		
EP 568681	[illegible]	[illegible]	EP [illegible]	[illegible]	19[illegible]1119	199747	E
			WO [illegible]	A	19921119		
			EP [illegible]971057[illegible]	A	19921119		
DE 69222844	[illegible]	[illegible]	DE [illegible]		19921119	199802	E
			EP [illegible]92925384	[illegible]	19921119		
			WO [illegible]92US9910	[illegible]	19921119		
EP 798919	[illegible]	[illegible]	EP [illegible]5384	[illegible]	[illegible]1119	199816	E
			EP [illegible]26	A	[illegible]1119		
US 5751846	[illegible]	19[illegible]0512	US 1991[illegible]3	[illegible]	19911121	199826	E
			US [illegible]		[illegible]0307		
EP 798919	B1	[illegible]	[illegible]		[illegible]119	200014	E
			EP [illegible]		[illegible]119		
DE 69230695	[illegible]	[illegible]	DE [illegible]		[illegible]119	200022	E
			EP [illegible]		[illegible]21119		

Priority Applicati[illegible] kind, [illegible]: US 1[illegible] A 19911121; US 1994207284 A [illegible]

Patent Details
Number Kind [illegible] Filing Notes
WO 1993011629 A [illegible]
National Designate[illegible] States,Original: JP
Regional Designated States,Original: [illegible] CH [illegible] DE ES FR **GB GR IE IT LU**
MC NL SE
EP 568681 A1 EN [illegible] PCT Application WO **1992US9910**
Based on [illegible] patent WO **1993011629**
Regional Designated States,Original: [illegible] FR **GB** [illegible]
US 5339368 A [illegible]
JP 7502154 [illegible] PCT Application WO **1992US9910**

Based on O[illegible] patent WO 1993011629
EP 798919 A2 [illegible] 9 Division of application EP
1992925284

Division of patent EP 568681
Regional Designated States,Original: DE [illegible] GB [illegible]
EP 568681 B1 EN 3[illegible] PCT Application WO 1992US9910
Related to application EP
1997105726

Related to patent EP 798919
Based on OI patent WO 1993011629
Regional Designated States,Original: DE FR GB IT
DE 69222844 E DE Application EP 1992925284
PCT Application WO 1992US9910
Based on OI patent EP 568681
Based on OI patent WO 1993011629
EP 798919 A3 EN Division of application EP
1992925284

Division of patent EP 568681
US 5751846 A EN Division of application US
1991796703

Division of patent US 5339368
EP 798919 B1 EN Division of application EP
1992925284

Division of patent EP 568681
Regional Designated States,Original: DE [illegible] GB [illegible]
DE 69230695 [illegible] DE Application EP 1997105726
Based on OI patent EP 798919

Alerting Abstract W[illegible]

The method uses a packet size estimate [illegible] establish the packet size of the memory space required to store the image data for a given document after compression by discrete transform technique. The data comprises pixels each representing one of a number of grey levels.

A selection processor selects, as a function of the estimate, one of a number of transform coefficient modifier matrices, e.g. a matrix of quantising values. The selected matrix [illegible] is transmitted to a transform compressor [illegible] use in altering, e.g. quantising, the number of transform coefficients.

ADVANTAGE - Modified compression characteristics in real time. Maximises image quality and [illegible] characteristics.

Equivalent Alerting Abstract US A

The discrete transform image data compression system has frequency transform coefficients modified [illegible] matrix of quantiser values. The system uses a predefined number of quantization matrices to adaptively select, on [illegible] document-[illegible] approximate memory packet size for each document's compressed image [illegible] storage by

selecting
one of the number of quantization matrices in accordance with the
packet
size estimate obtained for each document image.
Additionally, the system generates contrast reduction and gray level
stretch remapping curves as a function of global image data
characteristics, which are key line [illegible] of the document image
data.
The remapping curves are utilized to preprocess the image data for more
effective data compression.
ADVANTAGE - Compression characteristics can be modified in real
time on
per document basis.

Title Terms/Index Terms/Additional Words: DOCUMENT; IMAGE; COMPRESS;
METHOD
; ESTABLISH; REQUIRE; MEMORY ; SPACE; SELECT; QUANTUM; MATRIX;
DEPEND;
NEED; STORAGE; DATA; AFTER; DISCRETE; TRANSFORM; TECHNIQUE
Class Codes
International Classification (Main): [illegible], H04N-001/46
International Classification (+ Attributes)
IPC + Level Value Position Status Version

G06T-0009/00	A	I	R	20060101
H04N-0001/407	A	I	R	20060101
H04N-0001/41	A	I	R	20060101
H04N-0007/26	A	I	R	20060101
H04N-0007/30	A	I	R	20060101
G06T-0009/00	C	I	R	20060101
H04N-0001/407	C	I	R	20060101
H04N-0001/41	C	I	R	20060101
H04N-0007/26	C	I	R	20060101
H04N-0007/30	C	I	R	20060101

US Classification, Issued: 382256000, 382041000, 3483 [illegible]000, 382169000,
382172000

File Segment: EPI
DWPI Class: T01; [illegible];
Manual Codes (EPI/S-X): [illegible]; T01-J[illegible]; T0[illegible]A1; T01-J10B; T01-J10B1;
T05-K02; W02-J0[illegible]
Document image compression method establishing required memory
space...

...selects quantizing matrix in dependence of memory space needed to
store image data after compression by discrete transform technique.

Alerting Abstract ...The method uses a packet size estimator to
establish
the packet size of the memory space required to store the image
data
for a given document after compression by discrete transform
technique...

...A selection process [illegible], as a function of the estimate, one
of a
number of transition [illegible] modifier matrices, [illegible] a matrix of

quantising values. Th[illegible] s[illegible] matrix of modifiers is transmitted to a
transform compressor for use in altering, e.g. quantising, the number...

...ADVANTAGE - Modifies compression characteristics in real time. Maximises image quality and background noise characteristics.

Equivalent Alerting Abstract ...The discrete transform image data compression system has [illegible] transform coefficients modified in accordance with a matrix [illegible] quantiser values. The system uses a predefined
number of quantization matrices to adaptively select, on a
document-by-document basis, an approximate memory packet size for each
document's compressed image data storage by selecting one of the...

...Additionally, the system generates contrast reduction and gray level stretch remapping curves as a function of global image data characteristics, such as gray level histogram of the document image...

...ADVANTAGE - Compression characteristics can be modified in real time
on per document basis.

Title Terms.../Index Terms/Additional Words: MEMORY ;

Original Publication Data by Authority

Original Abstracts:
A discrete transform image data compression system in which frequency transform coefficients are modified in accordance with a matrix of quantizer values employs a predefined plurality of quantization matrices to
adaptively select, on a document-by-document basis, an approximate memory
packet size for each document's compressed image data storage by selecting
one of the...
...the system employs generation of contrast reduction and gray level stretch remapping curves as a function of global image data characteristics, such as gray level histogram of the document image...

...A discrete transform [illegible] compression system (Figure 2) in which
frequency transform coefficients are modified in accordance with a matrix
of quantizer values [illegible] predefined plurality of quantization matrices
to adaptively select, on a document-by-document basis, an approximate memory packet size for each document's compressed image data storage by
selecting one of the...

...the system employs generation of contrast reduction and gray level stretch remapping curves as a function of global image data characteristics, such as a gray level histogram of the document image...A discrete transform image data compression system in which frequency transform coefficients are modified in accordance with a matrix of quantizer values employs a predefined plurality of quantization matrices to adaptively select, on a document-by-document basis, an approximate memory packet size for each document's compressed image data storage by selecting one of the...

...the system employs generation of contrast reduction and gray level stretch remapping curves as a function of global image data characteristics, such as a gray level histogram of the document image...

...A discrete transform image data compression system in which frequency transform coefficients are modified in accordance with a matrix of quantizer values employs a predefined plurality of quantization matrices to adaptively select, on a document-by-document basis, an approximate memory packet size for each document's compressed image data storage by selecting one of the...the system employs generation of contrast reduction and gray level stretch remapping curves as a function of global image data characteristics, such as a gray level histogram of the document image...

...A discrete transform image data compression system in which frequency transform coefficients are modified in accordance with a matrix of quantizer values employs a predefined plurality of quantization matrices to adaptively select, on a document-by-document basis, an approximate memory packet size for each document's compressed image data storage by selecting one of the...

...the system employs generation of contrast reduction and gray level stretch remapping curves as a function of global image data characteristics, such as a gray level histogram of the document image...

Claims:

The method uses a [illegible] to establish the packet size of the memory space required to store the image data for a given document after compression [illegible] transform [illegible]...

...A selection processor selects, as a function of the estimate, one of a

number of transform [illegible] modifier [illegible] ices, e. [illegible] a matrix of quantising values. T[illegible]ed matrix [illegible] fiers [illegible] transmitted to a transform compressor for use in altering, [illegible] antising, the number...

...der Transforma[illegible]essor (119) [illegible] Transformationsko[illegible] verwendet, um di[illegible] zu bewi[illegible] das [illegible]stem zum Modifizieren der Transfo[illegible]ationsk[illegible] effizi[illegible] umfass[illegible]</br> a) eine Selektionsprozess[illegible]ng (1[illegible]) zum [illegible]rn mehrerer Modifizierer -Matrizen, wobei jed[illegible] Daten zum Ver[illegible]ern der Transformationskoeffizi[illegible] umfasst, wobei die Daten jeder Matrix... transform compress[illegible] utilizing [illegible] transform coefficients to effect compressio[illegible] said system for modifying the transform coefficients compris[illegible]</br> a, selecti[illegible] processor means (215) for storing a plurality o[illegible] modifier matrice[illegible] matrix of said plurality comprising data for [illegible] transform [illegible], the data of... ...size for a giv[illegible] image;</br> [illegible] estimating a packet size (213) of [illegible] required to [illegible] the [illegible] packet size (213), and operative to [illegible] of said plural[illegible] matri[illegible] as a function of a packet size [illegible] a function [illegible] desired [illegible] packet size;</br> d) s[illegible]) [illegible] operative to select...

...least one first matrix if the estimat[illegible] size is above the desired target packet size and operative to sele[illegible] least one second matrix if the estimated [illegible] is not above [illegible] target packet...gray level pixel data, [illegible] histogram [illegible]ng me[illegible] (107) further performing a stretch [illegible] conversion fu[illegible] of [illegible]verted highly visible gray level [illegible] and the nomin[illegible] visible gray level...

...and means (20[illegible] combining [illegible] and the gray level stretch conversion functions and for applying a combined conversion function to the document image data pri[illegible]ression to minimize loss of said nomin[illegible] level pixel [illegible] said histogram processing means (107) furth[illegible]ng a [illegible] conv[illegible] function of both the convert[illegible] the nominally visible gray... ...said histogram [illegible] trast reduction and the gray lev[illegible] and [illegible]pping means

(211) for applying the combined conversion functions to the document image data prior [illegible] data compression to [illegible] loss of said nominally visible...the transform compressor utilizing [illegible] plurality of transform coefficients to [illegible] compression, a system for modifying the transform coefficients comprising: means for storing [illegible] plurality of modifier matrices, each matrix of said plurality [illegible] data [illegible] altering transform coefficients, the data of...

...different packet [illegible] for a given document image; means for estimating a packet size of memory space required to store the document image data for a given document after compression by [illegible] discrete packet size, and operative to select one of said plurality of matrices as a function of a packet size estimate and [illegible] function [illegible] target packet size; said selection process [illegible] operative to select [illegible] first matrix if the estimated packet size is above the desired target packet size and operative to select at least one second matrix if the estimated packet size is not above the desired target packet [illegible].

...visible gray level pixel data, said histogram processing means further performing a stretch [illegible] conversion function of the converted highly visible gray level [illegible] data and the [illegible] visible gray level li means for combing the contrast reduction and the gray level stretch conversion functions and for [illegible] combined conversion functions to the document image data prior to data compression [illegible] minimize loss of said nominally v...

20/69,K/14 (Item 14 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0005785509 - Drawing available
WPI ACC NO: 1992-[illegible]
Related WPI Acc No: 1990-[illegible]965; 1990-25362[illegible]; 1990-267979; 1990-368343; 1991-036317; 1991-0[illegible]; 1991-2808[illegible]; [illegible]935; [illegible]-088529; 1993-093536; 199[illegible]; 1995-123120; [illegible]7305; 1996-230181; 1996-251121; 19[illegible]; 1997-033804; 199[illegible]65021; 1997-225521; 1997-271400; 1[illegible]; 1997-393070; [illegible]-5[illegible]789; 1997-549243; 1998-332816; 1998-52[illegible]; 1998-520722; [illegible]; 199[illegible]-131678; 1999-442814; [illegible]; 2000-052351; [illegible]; [illegible]-269746; 2000-586426; 20[illegible]; 200[illegible]-671740; [illegible]; [illegible]373508; 2002-279908; 2[illegible]
XRPX Acc No: N199[illegible]
Electronic key integrated circuit - contains password protected sub-keys in partitions of secure [illegible] and scratch-pad [illegible] for moving data into partitions
Patent Assignee: [illegible] SEMICONDUCTOR [illegible]; DALLAS SEMICONDUCTOR CORP (DALL-N)
Inventor: BOLAN M L; CURRY S J; CURRY S M; [illegible] K E; LEE R D; LEHMANN G H; LEO R D; PAYNE [illegible]
Patent Family ([illegible] patents, 17 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 19[illegible]019067	[illegible]		[illegible]	A	199[illegible]	199201	B
			[illegible]	[illegible]	1991		
US 5[illegible]05	[illegible]	[illegible]	US [illegible]	[illegible]	1989	199318	E
			US [illegible]		1989		
			US [illegible]		1989		
			[illegible]		1989		
			[illegible]		1989		
			US 1989[illegible]		1989		
			US 1989[illegible]		1989		
			[illegible]		1989		
			[illegible]		1989		
			[illegible]		1990		
US 5[illegible]946			[illegible]		1990	199320	
NCE							
US 5[illegible]	[illegible]		[illegible]		1989	199328	E
			[illegible]		1989		
			[illegible]		1989		
			US [illegible]		1989		
			[illegible]		1989		
			[illegible]		[illegible]		
			[illegible]		[illegible]		
			US [illegible]		[illegible]		
US 53[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	199416	E
			[illegible]		[illegible]		

			US 198935159	A	19890515		
			US [illegible]	A	[illegible]0515		
			US [illegible]	A	[illegible]0515		
			US [illegible]	A	[illegible]0515		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]		[illegible]		
US 5506991	A	[illegible]0409	US [illegible]	A	[illegible]	199620	E
			US [illegible]	A	[illegible]		
			[illegible]		[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	1989[illegible]		
			US [illegible]	A	1989[illegible]0515		
			US [illegible]	A	19890515		
			US [illegible]	A	[illegible]0515		
			US [illegible]	A	[illegible]0515		
			US [illegible]	A	[illegible]		
US 5603000	A	19970211	US [illegible]	A	[illegible]	199712	E
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
US 5787498	A	[illegible]0728	US [illegible]	A	[illegible]	199837	E
			US [illegible]	A	[illegible]		
			US 198935159	A	19890515		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
			US [illegible]	A	[illegible]		
US 5210846 NCE	B1	[illegible]0629	US [illegible]	A	[illegible]	199932	
US 6035282	A	20000307	US [illegible]	A	[illegible]	200019	E
			US [illegible]	A	[illegible]		
			[illegible]		[illegible]		

JP 3334717 B2 20021015 [illegible] A [illegible] 200275 E

Priority Applications (no., kind, [illegible]): [illegible] 19980507; US 1996695505 A 1996[illegible]; US 199[illegible] [illegible]992882244 A 19920508; WO 1990US2[illegible] A 1990[illegible]; [illegible] 19890515; US 1989352596 A 1989[illegible]515; US 19[illegible] [illegible] 1989352142 A 19890515; US 1989351[illegible] A 19[illegible] [illegible] [illegible]9890515; US 1989351997 A 1989[illegible]; US 19[illegible] [illegible]05[illegible] [illegible]989351759 A 19890515; US 199061[illegible] A 19[illegible] [illegible] 561[illegible] [illegible]9901119; US 1990615608 A 19901119; US 1[illegible] [illegible] [illegible]990631929 A 19901219

Patent Details

Number Kind Lan Pg Fig Filing [illegible]

WO 1991019067 A EN 43 28

National Designated States,Original: [illegible]

Regional Designated States,Original: AT [illegible] GB GR **IT LU NL** SE

Number	Kind	Lan	Pg	Fig	Filing Details
US 5206905	A	EN	28	25	C-I-P of application US 1989351759
					C-I-P of application US 1989351760
					C-I-P of application US 1989351997
					C-I-P of application US 1989351998
					C-I-P of application US 1989351999
					C-I-P of application US 1989352142
					C-I-P of application US 1989352581
					C-I-P of application US 1989352596
					C-I-P of application US 1989352598
					C-I-P of patent US 4[illegible]5217
					C-I-P of patent US 4[illegible]9954
					C-I-P of patent US [illegible]377
					C-I-P of patent US 4[illegible]371
					C-I-P of patent US 19[illegible]004
					C-I-P of patent US [illegible]675
					C-I-P of patent US [illegible]771
US 5226137	A	EN	27	25	C-I-P of application US 1989351759
					C-I-P of application US 1989351760
					C-I-P of application US 1989351997
					C-I-P of application US 1989351998
					C-I-P of application US 1989351999
					C-I-P of application US 1989352142
					C-I-P of application US 1989352581
					C-I-P of application US 1989352596
					C-I-P of application US 1989352598
					C-I-P of [illegible]217
					C-I-P of patent [illegible]
					C-I-P of [illegible]7
					C-I-P [illegible]371
					C-I-P of patent [illegible]004
					C-I-P of patent [illegible]675
					[illegible] patent [illegible]771
US 5306961	A	EN	28	25	[illegible] application [illegible] 1989351759
					[illegible] [illegible]989351760
					C-I-P [illegible] 1989351997
					C-I-P [illegible] 1989351998
					[illegible] P [illegible] 1989351999

					C-I-P of application US 1989352142
					C-I-P of application US 1989352581
					C-I-P of application US 1989352596
					C-I-P of application US 1989352598
					Continuation of application US
1990615618					
					C-I-P of patent US 49[illegible]5217
					C-I-P patent US 49[illegible]8954
					C-I-P patent US 49[illegible]2377
					C-I-P patent US 49[illegible]2371
					C-I-P patent US 49[illegible]5001
					C-I-P of patent US 50[illegible]5675
					C-I-P of patent US 50[illegible]1771
					C-I-P of patent US 52[illegible]0846
US 5506991	A	EN	30	29	C-I-P of application US 1989351759
					C-I-P of application US 1989351760
					C-I-P of application US 1989351997
					C-I-P of application US 1989351998
					C-I-P of application US 1989351999
					C-I-P of application US 1989352142
					C-I-P of application US 1989352581
					C-I-P of application US 1989352596
					C-I-P of application US 1989352598
					C-I-P of patent US 49[illegible]5217
					C-I-P of patent US 49[illegible]8954
					C-I-P of patent US 49[illegible]2377
					C-I-P of patent US 49[illegible]2371
					C-I-P of patent US [illegible]
					C-I-P of patent US 50[illegible]
					C-I-P of patent US [illegible]
					C-I-P of patent US [illegible]846
US 5603000	A	EN	30	25	C-I-P of application US 1989351759
					C-I-P of application US 1989351760
					C-I-P of application US 1989351997
					C-I-P of application US 1989351998
					C-I-P of application US 1989351999
					C-I-P of application US 1989352142
					C-I-P of application US 1989352581
					C-I-P of application US 1989352596
					C-I-P of application US 1989352598
					C-I-P of application 1990US2891
					Continuation of application US
1990615615					
					C-I-P [illegible]
					C-I-P [illegible]
					[illegible]
					[illegible]
					[illegible]
					[illegible]
					[illegible]
					[illegible]
US 5787498	A	EN			C-I-P of application US 1989351759
					C-I-P of application US 1989351760
					C-I-P of application US 1989351997
					C-I-P of application US 1989351998
					C-I-P of application US 1989351999
					C-I-P of application US 1989352142

ation 1989352581
a ic 1989352596
C-I-P ation US 1989352598
Continuation of application WO

1990US2891

Continuation of application US

1990615615

Continuation of application US

1994259290

C-I-P US 4 217
US 954
C-I-P US 4 77
C-I-P patent US 4 371
C-I-P US 4 004
C-I-P US 375
C-I-P of patent US 50 771
C-I-P patent US 5 846
Continuation of patent US 5603000

US 6035382 A EN

C-I-P 1989351759
C-I-P 1989351760
C-I-P 1989351997
C-I-P 1989351998
C-I-P 1989351999
C-I-P 1989352142
C-I-P 1989352581
C-I-P 1989352596
C-I-P 1989352598
C-I-P ation 1990US2891
US

1990615615

Continuation of application US

1994259290

Continuation of application US

1996695505

C-I-P 3217
954
377
271
49
175
71
846
US 5603000
Continuation of patent US 5787498

JP 3334717 B2 JA 15

WO 438
JP

06502645

1992008725

Alerting Abstract WO A

The integrated circuit electronic containing multiple low-power memory cells. ers are connected to receive incoming comma the memory array. Access requests are translate any given

time multiple partitions of the array are assigned as secure memories which are read accessible.

A scratchpad memory is defined in one partition of memory. Data in the scratchpad memory can be moved to another part of memory if a block move request, accompanied by a correct password is decoded.

USE/ADVANTAGE - In low power electronic [illegible] integrated circuits containing secure data. Returns same data [illegible] possible passwords. Low power consumption. [illegible] various types of nonvolatile modules. Highly secure.

Equivalent Alerting Abstract US A

The electronic key device has a memory [illegible] pseudo-random number generator, connected to receive a seed value and to output a number which is strictly dependent on the seed value, but which is a nonlinear and non-monotone function of the seed value. External connections receive a password, and outputting data.

A digital comparator, connected to compare [illegible] password with a store value, enables output of data from the memory when the password does match the stored value and enables [illegible] data from the psuedo-random number generator when the password does not match the stored values. The pseudo-random number generator [illegible] connected to receive the password and to use the password as [illegible] a unique set of data is output from the generator [illegible] identical password.

ADVANTAGE - Electronic key hardware module [illegible] readily [illegible].

Equivalent Alerting Abstract US A

The integrated circuit electronic key has [illegible] array containing multiple low-power memory cells. Command [illegible] decoders are connected to receive incoming command [illegible] the memory array. Access requests are translated [illegible] that at any given time multiple partitions of the array are designed as secure memories which are read accessible.

A scratchpad memory is defined in one [illegible] of memory. Data in the scratchpad memory can be moved [illegible] memory if a block move request, accompanied by a correct [illegible].

USE/ADVANTAGE - [illegible] circuits containing secure [illegible] possible passwords. Low power consumption. [illegible] various types of nonvolatile modules. Highly secure.

Equivalent Alerting Abstract US A

The electronic key integrated circuit [illegible] independently addressable partitions of secure memory [illegible] three partitions

can function as a separate "subkey". Each of the subkeys is independently password-protected.

In addition to the secure subkey [illegible], the integrated circuit also contains a read/write "[illegible]" [illegible], which is the same size as each of the subkeys. After data has been written into the scratchpad (and verified if desired), it can be copied, as a block, onto one of the subkey partitions. However, to perform such a block move the password of the target subkey must also be [illegible].

ADVANTAGE - Provides high degree of security.

Equivalent Alerting Abstract US A

The integrated circuit can be [illegible] in a battery-backed mode or in a battery-[illegible] mode. The [illegible] pin (BAT) for battery input, and a [illegible] (VCCI) [illegible] power supply.

Two PMOS switches are provided to connect [illegible] these two pins to an on-chip power supply [illegible] under appropriate [illegible].

The switches are P-channel insulated gate FETs with a width-to-length ratio greater than 20[illegible]. A third pin can alternatively be used for battery input. The logic which controls the [illegible] ensures that they will not turn on if their respective [illegible] input [illegible] BAT are low.

ADVANTAGE - [illegible] require [illegible], laser writing or fuse blowing.

Title Terms/Index Terms/Additional Words: [illegible]; INTEGRATE; CIRCUIT; CONTAIN; PASSWORD; PROTECT; SU[illegible]; MEMORY ;

SCRATCH; PAD; AREA; [illegible]; DATA

Class Codes

International Classification (Main): G07F-[illegible], G06F-012/00, G06F-012/02,

G06F-012/14, G06F-013/00, G06F-0[illegible], [illegible], H03K-003/01

(Additional/S[illegible]): A[illegible] [illegible]/6515,

G06K-[illegible]/06, [illegible]

US Classification [illegible]: [illegible]95725000,

364D[illegible], [illegible], 3642[illegible] [illegible]25000,

395400000, [illegible] [illegible]25000,

364D[illegible] [illegible]06400,

30735[illegible]00, 307[illegible] 229 [illegible]29500,

3642[illegible] [illegible] 1211 [illegible]4000

File Segment: E[illegible];

DWPI C[illegible]: T01; [illegible]

Manual [illegible]: T01 [illegible] 1C[illegible]

...contains password protected [illegible] memory and scratch-pad [illegible]

Original Titles:

...Integrated circuit memory with verific[illegible] unit which resets an address translation register upon failure [illegible] one-to-one correspondences between addresses and memory [illegible]s...

...Integrated circuit memory with verific[illegible] unit which resets an address translation register upon failure [illegible] one-to-one correspondences between addresses and memory [illegible]s...

...Circuit for [illegible] a [illegible] word [illegible] subkey...

Alerting Abstract ...The integrated circuit electronic [illegible] has a memory array containing multiple low-power memory [illegible]. Command and address decoders are connected to receive incoming [illegible] requesting access to the memory array. Access requests are translated in such a pattern that at any given time multiple partitions of the array are assigned as secure memories which [illegible] and accessible...

...A scratchpad memory is defined in one [illegible] of memory. Data in the scratchpad memory can be moved to another [illegible] memory if a block move request, accompanied by a correct password [illegible]...

Equivalent Alerting Abstract ...The electronic [illegible] device has a memory an a pseudo-random number generator, connected to receive a seed value and to output...

...is strictly dependent on the seed value, [illegible] which is a nonlinear and non-monotone function of the seed value. [illegible] connections receive a password, and [illegible] data...

...to compare [illegible] password [illegible], enables output of data from the memory when the password [illegible] the stored value and enables output [illegible] from the...

...The integrated [illegible] key [illegible] containing multiple low-power memory [illegible]. Command [illegible] address decoders are connected to [illegible] incoming [illegible] requesting access to the memory array. Access [illegible] are translated in such a pattern that at any given time multiple partitions of the array are [illegible] memories which [illegible]...

...A scratchpad memory is defined in one partition of memory . Data in
the scratchpad memory can be moved to another part of memory if a block
move request, accompanied by a correct password is decoded...

...The electronic [illegible] integrated circuit in[illegible] three independently
addressable partitions of secure memory . [illegible] of these three partitions
can function as [illegible] "subkey" . Each [illegible] subkey is independently
password-protected...

...In addition to the secure subkey memory partitions, the integrated
circuit also contains a read/write "scratchpad" memory , which is the
same size as each of the subkeys. After data has been written into the
scratchpad [illegible] verified...

Title Terms.../Index Terms/Additional Words: MEMORY ;

Original Publication Data by Authority

Original Abstract:

...pseudo-random number generator. If the [illegible] password is received, the
contents of [illegible] will be output [illegible] electronic key.
However, if an incorrect password is received [illegible] will be used
...

...An electronic [illegible] integrated circuit which includes three independently
addressable partitions of secure memory . Each of these three partitions
can function as [illegible] separate "subkey." Each [illegible] is
independently password protected. In addition [illegible] the secure subkey
memory partitions, the integrated circuit [illegible] contains a read/write
"scratchpad" memory , which is the same size as each of the
subkeys.
After data has been written into the scratchpad (and verified if
desired), it can [illegible] as a block, [illegible] the [illegible]key...

...A low-power secure memory in which block [illegible] operations are
performed
without [illegible] transfer register holds a set of
pointers [illegible] values in
this [illegible] SRAM cell
locations in the memory array [illegible] without
performing any [illegible] operations in the array [illegible] the charge
consumption which would [illegible] required [illegible] charging and discharging
bitlines [illegible] read [illegible] chip of

the
preferred embodiment includes a scratchpad memory as well as multiple secure memories [illegible]. The [illegible] Block command can transfer a block of data from the scratchpad memory directly into the corresponding block location within a secure subkey, or can replace the entire contents of the secure [illegible] partition (including the ID and Password fields) with the entire contents of the scratch...

...A low-power secure memory in which block [illegible] operations are performed without extensive write operations. A translation register holds...

...pointers which affect the address decoding. By changing the values in this special register, the logical address [illegible] physical SRAM cell location in the [illegible] array [illegible] changed without performing any write operation [illegible] ...

...the charge consumption which [illegible] required for charging and discharging bitlines as the memory cells are read and written to.

The chip of the preferred embodiment includes a scratchpad memory as well as multiple secure memories (multiple "[illegible]"). The Move Block command can transfer a block of data from the scratchpad directly into the corresponding block location within a secure subkey, or can replace the entire contents of [illegible] (including the ID and Password fields) [illegible] scratchpad. For security purposes, the...

...An integrated circuit with a secure memory location is comprised of a memory and a circuit which receives a twenty-four bit command word. At least one location in the memory stores [illegible] the secure subkey.

The circuit responds to the command which corresponds to a secure memory location [illegible] a starting address [illegible] transmitted in both true and bit complement form. Each [illegible] of a 64-bit ID field, a 64-bit password field and [illegible] field.

...

...An electronic key integrated circuit [illegible] independently addressable partitions [illegible] memory. [illegible] partitions can function as a separate "[illegible] key". Each of [illegible] independently

password-protected. In add[illegible] to the sec[illegible] memory partitions,
the integrated circuit also contains [illegible] read [illegible] "[illegible]" memory ,
which is the same size as each [illegible] [illegible] data has been
written into the scratchpad [illegible] verified [illegible], it can be copied, as
a block, onto one of the [illegible] partitio[illegible], to perform such a
block move the password [illegible] the [illegible] be specified.

Claims:

An electronic key device, [illegible] pseudo-random number
generator, connected to receive a seed val[illegible] and to output a number which
is strictly dependent on said seed value, [illegible] which is [illegible] nonlinear and
non-monotone function of said seed val[illegible] [illegible] connections for
receiving a password, and external connect[illegible] for outputting data;
a...

...received password with a st[illegible] value, [illegible] enabl[illegible] output of data
from said memory wh[illegible] said [illegible] [illegible] red value and
(2) to enable output of da[illegible]...

...including: exactly two [illegible] [illegible] portions accessible
on the exterior of said each module; memo[illegible] input logic, operatively
connected to first and second [illegible] [illegible] portions, and
configured to detect whether s[illegible] first [illegible]...

...said second voltage, depending on the [illegible] data bit being read
from said memory , a[illegible] th[illegible] said [illegible] after a
predetermined delay, determined by said tim[illegible] has
elapsed; in
said write...

...then test the voltage of [illegible] portion [illegible] data bit in
said memory according to [illegible]...

...

...An integrated circuit, comprising: [illegible] memory array
containing multiple low-power memory cells arranged in rows and
columns; a command decoder, connected [illegible] requesting
access to said memory ar[illegible], [illegible] decoder includes a
translation register which [illegible] correspondences
between logical memory a[illegible] cell[illegible] with each of
said one-to-one corre[illegible] [illegible] bit
pattern in said register; [illegible] translates

access requests, in accordance with the bit pattern of said translation register, to provide a block select output [illegible] address [illegible], connected to...

...to select ones of said rows and columns [illegible] said cells for access; wherein said command and address decoder [illegible] access requests in such a pattern that, multiple partitions [illegible] said [illegible] assigned as secure subkey memories , which are only accessible with correct password, each partition having an independent password and at least one partition of said array is assigned as scratchpad memory , which is accessible without password protection; and [illegible] in one of the access requests which said command decoder can recognize is a block move, to secure subkey memory , request, and in response thereto said bit pattern in said translation register is altered to effect the requested block move by changing a plurality of...

...if the block move request is not [illegible] the [illegible] password for the secure subkey memory portion which would be altered by the requested block move .

...Claim 9. An integrated circuit, comprising: at least one memory array containing multiple low-power memory cells arranged in rows and columns; a command decoder, which command decoder [illegible] a translation register having modifiable contents , which translation register defines plural one -to-one correspondences between logical memory addresses and cells of said array, with [illegible] correspondences associated with a corresponding bit pattern in said [illegible]; andwherein said command decoder also includes verification means which monitors said translation register, and if the bit pattern in said translation register at any time ceases to define a one-to-one correspondence between said logical memory addresses and [illegible] on means for [illegible]

...

...An integrated circuit, comprising: at least one memory array containing multiple low- power memory cells [illegible], a command decoder, connected to decode [illegible] said memory array, wherein said command decoder [illegible] a [illegible] translation register, [illegible] contents, wherein said [illegible] plural

one-to-one correspondences between logical memory addresses and the
cells of said memory array, with each of the one-to-one
correspondences associated with a respective bit pattern in said
writable translation register; and wherein said command decoder
translates access requests, in accordance with the bit pattern of said
translation register, to provide a block select output; an address
decoder, connected to receive said block select output, and accordingly to
select ones of said rows and columns of memory cells; and wherein said
command decoder further screens access requests for match with a password,
and further comprising a pseudo-random number generator, which is activated
if said command decoder does not detect a match with the password;
andmeans to monitor the integrity of said translation register
and for providing a reset signal if it detects an inconsistency
therein.

20/69,K/15 (Item 15 from file: 350) **Double Patenting?

DIALOG(R)File 350:Derwent WPIX

0005464644 - Drawing available

WPI ACC NO: 1991-067411/199110

XRPX Acc No: N1991-052152

Processing prolog object words in computer memory - by testing arbitrary word in address using remaining bits as pointer

Patent Assignee: IBM CORP (IBMC); [illegible] (IBMC)

Inventor: GILLET M J; GILLET [illegible]

Patent Family (4 patents, [illegible] countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 415894	A	19910306	EP 1990850250	[illegible]	19900626	199110	B
US 5386520	A	19950131	US 1989393629	[illegible]	19890814	199511	E
			US [illegible]	A	[illegible]		
EP 415894	B1	[illegible]	EP 1990850250	[illegible]	[illegible]	199550	E
DE 690[illegible]576	[illegible]	[illegible]	DE 690[illegible]	[illegible]	19900626	199605	E
			EP 199[illegible]	[illegible]	19900626		

Priority Applications (no., kind, date): [illegible] 199[illegible]28315 A 19920827; US 1989393629 A 19890814

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes	
EP 415894	A	EN				
Regional Designated States,Original: DE FR GB						
US 5386520	A	EN	42	37	Continuation of application	US 1989393629
EP 415894	B1	EN	55			
Regional Designated States,Original: DE FR GB						
DE 69[illegible]576	E	DE			Application EP 1990850250	
					Based on OPI patent EP 415894	

Alerting Abstract EP A

The technique includes testing a bit of an arbitrary word stored in **memory** to determine whether it is a pointer [illegible]. The remaining bits of the word are used as a pointer to an address in the **memory**.

This occurs if the bit is set to a value representing an object word. The object words are processed in the memory as non-typed pointers.

ADVANTAGE - Improves overall performance. @(59pp Dwg.No.5/37)@

Equivalent Alerting Abstract US A

The computer system [illegible] method involves [illegible] using the processor of the computer system, [illegible] instructions which [illegible] then stored in the memory [illegible] instructions including object word [illegible] type [illegible] All [illegible] words of type pointer [illegible] the most significant bit of the object word [illegible] single

address field. All object words of type descriptor include a tag field in the most significant position of the object word, the tag field having the most significant bit set one one.

The step of encoding a predicate further involves storing a type pointer in memory having an object address which points to itself and is representative of a free variable, and executing using the processor of the computer system, the set of instructions. The executing step further involves determining using the processor of the computer system, the type of object word present. The determining step involves loading an object word into a register having the same number of bits as the object word, using the object word of type pointer as an address of an object if the most significant bit in the register is equal to zero, and using the object word as a type descriptor having a tag field if the most significant bit in the register is equal to one.

ADVANTAGE - Improved addressing and increase in effective address spaces of Prolog interpreter/compiler are accomplished by allowing Prolog to efficiently employ address space of N-1 bits on computer whose words are made up of N bits.

Title Terms/Index Terms/Additional Words: [illegible]; OBJECT; WORD; COMPUTER; MEMORY; [illegible]; POINT

Class Codes

International Classification [illegible]

(Additional/Secondary): G06F [illegible]

US Classification, Issued: [illegible], 364DIG001

File Segment: EPI;

DWPI Class: T01

Manual Codes (EPI/S-X): T01-[illegible]; T01-F05

Processing prolog object words in computer memory [illegible]

Alerting Abstract ...The technique includes [illegible] of an arbitrary word stored in memory to determine whether it is a pointer or a descriptor. The remaining bits [illegible] pointer to an address in the memory.

...

...set to a value representing an object [illegible] are processed in the memory [illegible]

Equivalent Alerting Abstract ...The computer system operating method involves encoding, using the processor of the computer system, a predicate into a set of instructions which are stored in the memory,

the
instructions including [illegible] type
desc[illegible].
All object words [illegible] rise...

...The step of enc[illegible] a predicate [illegible] ring a type
pointer in memory having an object address which [illegible] to itself and
is
representative of a free variable, and executing using the processor of
the
computer system, the set of instructions. The [illegible] step further
involves determining using the processor of the computer system, the
type
...

...[illegible] The determining step involves [illegible] an object word
into
a register having the same [illegible] of [illegible] word,
using
the [illegible] word of type pointer as an [illegible]...

Title Terms.../Index Terms/Additional Words: MEMORY ;

Original Publication Data by Authority

Claims:
1. Method for efficiently [illegible] in an
arbitrary word in a computer memory [illegible] of: (a)
testing [illegible] to
determine
if it is a pointer [illegible] bits
of
the arbitrary word [illegible] memory, if the bit
is
set to a value indicative of an object word...

...A method for processing a prolog object contained in an arbitrary
word
in a computer memory, [illegible] steps of (a)
testing a single bit of the [illegible] single bit
is
set to a first pre[illegible]

...[illegible] bits [illegible] dress in
the
memory if the single bit [illegible] value
(30).
...

...A [illegible] or, a
memory
and [illegible] from a Prolog
pro[illegible] nd
including
object words, comprising [illegible] processor
of

the computer system, an object word into a register having the same
number of bits as the object word; using the processor of the computer
system to process the object word [illegible] type [illegible] as an address of an
object having no tag field and having [illegible] address field

20/69,K/16 (Item 16 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0003228161
WPI ACC NO: 1984-08[illegible]3/[illegible]415
Image processor for photocopies, facsimile transmitters etc. - has processing signal indicat[illegible] with associated [illegible] and required signal
memory
Patent Assignee: CANON KK (CANO)
Inventor: ARIMOTO S; SHIMIZU K; SHIMIZU K K; YAMADA M; YAMADA M K
Patent Family (21 patents, 4 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
DE 33[illegible]	A	[illegible]	DE 33[illegible]657	A	19[illegible]30[illegible]	198415	B
JP 5[illegible]	A	[illegible]	JP [illegible]		[illegible]	198422	E
			JP [illegible]218[illegible]		19[illegible]21013		
			JP [illegible]218[illegible]		1982[illegible]25		
			JP [illegible]		[illegible]		
GB 213[illegible]	[illegible]	[illegible]	GB [illegible]		[illegible]	198423	E
JP 59075[illegible]	A	1984[illegible]	JP 19[illegible]172435		19821001	198423	E
			JP [illegible]2182[illegible]		[illegible]		
			JP [illegible]		[illegible]21[illegible]		
JP 59075755	A	19840[illegible]3	JP 198217253[illegible]	A	19821[illegible]	198423	E
			JP 19[illegible]218[illegible]		[illegible]21[illegible]		
			JP [illegible]		[illegible]210		
GB 2130857	B	1987[illegible]				198716	E
DE 333565[illegible]	[illegible]	19[illegible]	DE [illegible]		[illegible]30[illegible]	199001	E
DE 33[illegible]	A	[illegible]	DE [illegible]		[illegible]	199142	E
JP 5[illegible]	A	19[illegible]	JP [illegible]		[illegible]	199346	E
			JP [illegible]		[illegible]		
JP 5[illegible]	A	19931015	JP [illegible]		[illegible]	199346	E
			JP [illegible]		[illegible]		
JP 5265[illegible]	A	[illegible]	JP [illegible]		[illegible]	199346	E
			JP [illegible]		[illegible]		
DE 3348475	A1	[illegible]	DE [illegible]		[illegible]	199347	E
			DE 3[illegible]471	A	19[illegible]		
DE 3348[illegible]	A1	1994[illegible]	DE 333[illegible]		[illegible]30[illegible]	199406	E
			DE [illegible]		[illegible]30		
US 5369733	A	1994[illegible]	US [illegible]		19[illegible]	199502	E
			US [illegible]		[illegible]		
			US [illegible]		[illegible]		
JP [illegible]		[illegible]	JP [illegible]		[illegible]	199511	E
			JP [illegible]				
DE 33[illegible]	C2	19[illegible]	DE [illegible]		[illegible]	199623	E
			DE [illegible]		[illegible]		
DE 3[illegible]	[illegible]	[illegible]	DE [illegible]		[illegible]	199945	E
			DE [illegible]		[illegible]		
US 5377[illegible]	A	19[illegible]	US [illegible]		[illegible]	199953	E
			US [illegible]		[illegible]		
			US [illegible]		[illegible]		
			US 19[illegible]	A	[illegible]		
			US [illegible]				
DE 33[illegible]	[illegible]	[illegible]	DE [illegible]		[illegible]	200004	E
US 6[illegible]	[illegible]	[illegible]	US [illegible]		[illegible]	200165	E

			US		
			US		
			US		
			US		
US 6329979	B1	20011211	US [illegible]	[illegible]	200204 E
			US [illegible]		
			US [illegible]		
			US [illegible] 21721	A 1993[illegible]	
			US [illegible]		
			US [illegible]		

Priority Applications [illegible] 19821018; JP 1983[illegible] A [illegible] 19821018; JP [illegible] 19821018; JP [illegible] A [illegible]

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
DE 3335657	A	DE	103	18	
JP 5265290	A	JA			Division of application JP 1982172935
JP 5265291	A	JA			Division of application JP 1982172935
JP 5265292	A	JA			Division of application JP 1982172935
DE 3348[illegible]	A1	DE			Division of application DE 3335657; Division of patent DE 3335657
DE 3348476	A1	DE			Division of application DE 3335657; Division of patent DE 3335657
US 5369753	A	EN	57	18	Continuation of application US 1983536982; Continuation of application US 1986911069
JP 7005958	A	JA	13		Division of application JP 1982[illegible]
DE 33[illegible]	C2	DE	59	15	Division of application DE 3335657; Division of patent DE 3335657
DE 3348[illegible]	C2	DE			Division of application DE 3335657; Division of patent DE 3335657
US 5[illegible]	A	EN			Continuation of application US 1983536982; Continuation of application US 19[illegible]538; Continuation of application US 19[illegible]310; Continuation of application US 19[illegible]21
DE 3348[illegible]	C2	DE			Division of application DE 3335657

			Division of patent DE 3335657
US 6307540	B1	EN	Continuation of application US 1983536982
			Continuation of application US 1987110538
			Continuation of application US 1990[illegible]310
			Continuation of application US 19[illegible]
			Division of application US 1994[illegible]
			[illegible] 5977954
US 6329970	B1	EN	Continuation of application US 1983536982
			Continuation of application US 1987110538
			Continuation of application US 19[illegible]310
			Continuation of application US 19[illegible]721
			Division of application US 1994[illegible]
			Division of patent US 5977954

Alerting Abstract DE A

An indicator is provided for any a[illegible] signal information, relat[illegible] image processing. A keyboard is asso[illegible] the indicator, and a memory stores required information signals.

A control device operates the indicator [illegible] image transmission for information to the indicator. A second indicator may be provided for further information signals as requi[illegible] displayed by the first indicator. Further image processing [illegible] may be provided, while a memory stores such input signals. The sign[illegible] in the memory may be changed by a special device. The sys[illegible] and is suitable for facsimile equipment, p[illegible] rs a[illegible] processors.

Equivalent Alerting Abstract DE C

The image processing system has a [illegible] ering the image processing parameters. Each of the [illegible] used for several different functions. The [illegible] indicated by an associated display (252). The function [illegible] each operation of the function key (2?1) with the [illegible] param[illegible] enter[illegible] successive [illegible]

The [illegible] image processing para[illegible] number of stored and read out by operation of [illegible] each associated

with a respective store. Difference ... ered image processing parameters may be held in different storage locations.
USE - For selecting copy format ... copier.
USE - ... (pp)

Equivalent Alerting Abstract US A
The image processor system comprises ... which are mechanically and functionally separ... such ... that they can be used independently of each other and ... transmission of image information ... is possible in between. The reader includes an operating unit which cooperates with the printer to perform ... functions such as the image operation function, the image transfer function, the ... presetting function and the image quality processing function in addition to a function of a conventional copier. ... consists of a purpose ... portion, and a soft key portion which ... for optional creating the copy transfer functions and having function keys and displays corresponding to the keys.
The ... to a user in cooperation with a driver, CPU and ... A function key preset key portion is used for maintaining, reading and restoring the image transfer functions and having ... keys and preset key displays ... display the results ...
Title Terms/Index Terms/Additional Words: IMAGE ...; PHOTOCOPY; FACSIMILE; TRANSMIT; PROCESS; SIGNAL; INDICATE; ... STATE; KEYBOARD; ... MEMORY

Class Codes
International Classification (Main): ... 5/00, ...06F-0... /00, G06F-003/...
, ... 5/00, H04N-001/...
(Additional/Secondary): ...-003/16 ... 17/5 , ...-0.../00, G06F-015...
, ... H03M-..., ...3M- ... -001/387,
H... H04N-...
US ... 370081000,
... 0... 000

File ...
DWPI ...
Manu...

...h ... signal indicator ... keyboard, and requ...
sign...

Al... related to image ... tor , and a memory ...

...s display ... Further ... processing input ... be provided ... signals. The signal ... the memory ... changed by ... special ... The system enhan... user operation ... suitable ...

Equivalent Alerting Abstract ...processing system has an input keyboard
for ... the image processing parameters. Each of the function keys
(201) is used for several different functions ... function is
indicated by an associated display ... function is altered after
each operation of the function key ... image processing parameter ... function key ...

...The image processor system comprises ... and a reader which are
mechanically and functionally separated in such a manner that they can be
used independently of each other and ...

...between. The reader ... has ... unit which operates with the
printer to perform ... functions ...
the ... transfer function ... image
quality ... conventional ... The ... key portion,
and ... key portion which is used for ... the copy
transfer functions and having function keys and displays corresponding
to the keys...
...labels and messages to a user in ... operation ... a driver, CPU and
controller. A function key preset key portion is used for registering,
reading ... a standard
mode ... key, preset key and ... displays corresponding to...

Title ...

Original Publication Data ... Author...

Original Abstract:
An ... which are
mech... that they can
be ... image...

... between. The ... cooperates
with ...

operating
function, the image transfer function, the presetting function and the image quality processing function, in addition to a function of a communication ... The operating ... general purpose key ... creating the copy transfer functions ... having ... displays corresponding to ... key labels ... messages ... CPU and controller; and a function key ... portion which is used for registering, reading and resetting ... transfer functions and having a standard mode return key, ... keys ... key displays corresponding to the reset keys and...

...An image processing system comprises ... which are mechanically ... that they can be ... independently ... image information is ... operating unit ... such as the image ... function, the presetting function and the ... function, in addition to ... function ... operating unit consisting of a general purpose key portion; a soft key portion which is used for ... creating the copy ... functions and having function ... displays ... said displays being ... label ... in cooperation with ... key portion which is used for ... image transfer functions and ... and preset key displays corresponding to said reset keys and adapted to display the reset...

...An image ... comprising ... which are mechanically ... that they can ... image information ... operating unit ... such as the ... the presetting function and the image ... function,

in addition to a [illegible] function [illegible] of a conventional [illegible]. Said operating unit consists of a general [illegible] the key [illegible] a [illegible] of [illegible] portion which is used [illegible] the [illegible] functions and having function keys and displays [illegible] corresponding [illegible] said displays being [illegible] to display [illegible] label [illegible] in combination with a driver, CPU and controller; [illegible] key portion which is used for registering, reading and [illegible] the image transfer functions and having a [illegible] mode return key, preset keys and preset key displays [illegible] said [illegible] to display the [illegible] position...

...An image processing [illegible] comprises a [illegible] printer and a reader which are mechanically and functionally [illegible] that they can be used independently of each other and [illegible] of image information is possible therebetween. Said reader [illegible] an operating unit which cooperates with the printer [illegible] functions such as the [illegible] operating functions, the [illegible] transfer [illegible], the [illegible] of image [illegible] function, in addition to [illegible] functions of a conventional copier. Said operating unit consists of [illegible] key [illegible] soft key [illegible] which is used [illegible] the [illegible] functions and having function keys and displays corresponding [illegible] said displays being [illegible] to display [illegible] label [illegible] in combination with [illegible], CPU [illegible] controller; [illegible] key portion which is used for registering, reading [illegible] the image transfer [illegible] and [illegible] preset key displays corresponding [illegible] said [illegible] to display the re[illegible].

Claims [illegible]

...[illegible] which [illegible] portion which [illegible] operable to input different [illegible] in different stages of [illegible] instruction to [illegible] system [illegible] for displaying information [illegible] relating to the [illegible] input function of [illegible] for said p[illegible] ring

input
of [illegible] instruction to [illegible] said display [illegible] to [illegible] the displayed information [illegible] label [illegible] the current sub-instruction
for [illegible] manually operable portion [illegible] said information being
displayed in such a manner that it [illegible] corresponds to said manually operable portion.

[illegible] image processing system having an operating section which includes a manually operable portion which [illegible] different
sub-instructions in [illegible] instruction to the system
and [illegible] display means for displaying information which relates to
the [illegible] portion and which [illegible] label [illegible] provided a control [illegible] to cause said display means to change the displayed information [illegible] by the label to
indicate the current sub-instruction for [illegible] operable portion is operable, said information being displayed in such a manner that
it [illegible] corresponds to said manually operable portion.

An apparatus for [illegible] magnification [illegible] of a photocopying machine [illegible] apparatus [illegible] determination means for
determining [illegible] size of an image
being copied and a copy [illegible] size; [illegible] storing in respective memory locations [illegible] words indicative of a selected [illegible] the size of a reproduced image; a [illegible] selecting a
respective one of the memory locations; [illegible] control means for
reading the data word stored in the respective selected memory location
and [illegible] according to the
m[illegible] respective memory loc[illegible]
in[illegible] ratio in a
se[illegible]
one [illegible] memory
loc[illegible] means [illegible] ratio from
that [illegible] changing the
data [illegible] modified
data [illegible] ratio
in
the magnification [illegible]

[illegible] photographically
r[illegible] an original [illegible] on [illegible] recording material,

comprising a reading unit for optically reading the original image to be
reproduced on the sheet ... the ... display means having
a plurality of display positions ... display contents, and
for displaying plural ... information relating to one of i) a single image production operation to photo...

...corresponding to the plurality of ... positions of said display means ... input means ... additional
information, including ... for controlling reproduction
of the original image on the...

...corresponding to the plurality of positions of said plurality of input
means and being physically separate from said plurality of input means; control means for ... information relating to
the single image production operation to photo-electrically reproduce the
...

...sheet of the recording material ... said plurality of
input ... the ... corresponding to display ... responsive to a particular one of said plurality of input means
corresponding to the display position and for controlling displaying of
an additional message ... input of additional functional information from said ... means ... single image production operation ...

...the original image ... reading ... is executed
in accordance with ... from said plurality of input ... input ... information
from ... additional ... means for photo-electrically
transforming an image...

...read by said reading unit ... the recording material ... corresponding ... in accordance
with the input functional information ... additional functional information.

... a display unit
for ... input

units
is smaller than the number of parameters to be displayed on said display
unit, wherein said display unit is capable of changing the displayed contents and display the plural kinds of parameters for the information processing;memory means for storing a standard parameter group which includes plural kinds of parameters...

...er ... the single ... said reg...
mean...

...

...parameters as a parameter group necessary for executing one information
processing operation in response to a register instruction, said register
means registering the plural kinds of parameters using an operation button

...

...operation;display control means for controlling a display on the display
screen in accordance with the register instruction so that a guidance message information is displayed at a position on the...

...select a single one of said plural input means to which the parameter
group will be registered, wherein said display control means controls the
display on the display screen such that

20/69,K/17 (Item [illegible] from file: 350)
DIALOG(R)File 350:[illegible] WPI
(c) [illegible] The Thomson [illegible]

0003[illegible]248
WPI Acc NO: 1984-147821/[illegible]
Related WPI Acc No: 1987-356745; 1984-14781[illegible]; 1984-153991; 1984-172414; 1984-178193
Microcomputer with multiple-register proc[illegible] - has addressable memo[illegible]
locations and uses instructions having same bit size and format, with
reduced decoding dela[illegible]
Patent Assignee: EDWA[illegible] J (EDWA-[illegible]) [illegible] LTD (INM[illegible]); MAY M D (MAYM-[illegible])
ST[illegible] ELECTRONICS [illegible]
Inven[illegible]: [illegible]OWS J; [illegible] D; [illegible]
Patent Family (20 patent[illegible] [illegible])

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
EP 11[illegible]42	A	19840613	EP [illegible]33070[illegible]	A	[illegible]	198424	B
			EP 1[illegible]33070[illegible]	A	[illegible]		
			EP [illegible]	A	[illegible]		
			EP [illegible]330[illegible]	A	[illegible]		
			EP [illegible]307	A	[illegible]		
			EP [illegible]	A	[illegible]		
US [illegible]	A	[illegible]	US [illegible]	A	[illegible]	198730	E
DE [illegible]	G	[illegible]				198742	E
DE [illegible]	G	[illegible]				198743	E
US [illegible]	A	[illegible]	US [illegible]		[illegible]	198746	E
US [illegible]	A	[illegible]	[illegible]	A	[illegible]31[illegible]	198809	E
DE [illegible]	G	[illegible]				198815	E
DE [illegible]	G	[illegible]				198828	E
EP [illegible]	B	[illegible]	EP [illegible]		[illegible]	198838	
NCE							
			EP [illegible]		[illegible]		
			EP [illegible]				
			EP [illegible]		[illegible]		
			EP [illegible]33[illegible]				
			E[illegible] [illegible]720[illegible]				
DE [illegible]	G	[illegible]				198844	E
US [illegible]	A	[illegible]	US [illegible]659[illegible]	A	[illegible]	198916	E
US [illegible]	A	[illegible]	US [illegible]39[illegible]		[illegible]	199046	E
US [illegible]	A	[illegible]	[illegible]3[illegible]		[illegible]	199130	E
US [illegible]	A		[illegible]		[illegible]	199337	E
			[illegible] 1[illegible]338		[illegible]31[illegible]9		
			[illegible]		[illegible]		
JP [illegible]51	A	[illegible]	JP [illegible]2		[illegible]	199442	E
			JP [illegible]23[illegible]		[illegible]		
US [illegible]	A	[illegible]	US [illegible]35[illegible]		[illegible]	199543	E
			[illegible]				
			US [illegible]360		30		
			[illegible]360		27		
US [illegible]	A	[illegible]	[illegible]355		[illegible]	199612	E
			[illegible]693	A	[illegible]		
			[illegible]060				
			[illegible]36[illegible]				

US 5[illegible] A [illegible] [illegible] A [illegible]116 199620 E
US [illegible] A [illegible]
US [illegible] A [illegible]30
[illegible] A [illegible]
US 64[illegible] B1 [illegible]2078[illegible] US [illegible]3573 A [illegible]16 200248 E
US [illegible]693[illegible] A [illegible]209
US [illegible]060[illegible] A [illegible]030
US [illegible]366[illegible] A [illegible]024
US [illegible]547[illegible] A [illegible]507
US [illegible]833[illegible] A [illegible]303
US 2[illegible]54[illegible] A1 [illegible] US [illegible]35[illegible] [illegible]16 [illegible]0316 E
US [illegible]693[illegible] A [illegible]1209
US [illegible]60606[illegible] A [illegible]31030
US [illegible] [illegible]24
US [illegible] A [illegible]07
US [illegible] A [illegible]03
US [illegible] A [illegible]

Priority Applications (No., Kind, Date): [illegible]233[illegible] [illegible] 19821126

Patent Details

Number Kind Lan Pg Dw Filing Notes
EP [illegible] A [illegible] 107 [illegible]
Regional Designated States,Original: DE [illegible] GB IT NL [illegible]
EP [illegible] B [illegible]
Regional Designated States,Original: DE FR GB IT NL [illegible]
US 5[illegible] A [illegible] 41 18 Continuation of application US 19[illegible]3027
Division of application US 1986[illegible]8380
Division of patent US 4967326
JP 6[illegible] A JA 45 Division of application JP 1983[illegible]
US 5[illegible] A [illegible] 41 18 Continuation of application US [illegible]3027
Division of application US 198[illegible]
Division of application US 1990[illegible]
Division of patent US 4967326
Division of patent US 5243698
US 5491359 A EN 42 18 Continuation of application US 19[illegible]3027
Division of application US 198[illegible]
Division of application US 199[illegible]
Division of patent US 4967326
Division of patent US 5243698
US 5[illegible] A [illegible] [illegible] [illegible] Continuation of application US 19[illegible]3027

			Divisi . of application US
19861 18380			
			Divisi of appli ation US
1990C			
			Divi. of pat US 4967326
			Divisi of pa US 5243698
US 64 3 3	B1	EN	Contin of a lication US
19 53027			
			Divis of appl tion US
1986938380			
			Divi. of application US
19906			
			Contin ion of a plication US
19 366489			
			Contin of plication US
19 5 295			
			Division of patent US 4967326
			Divi. US 5243698
			Contin US 5491359
US 2 34544	A1	EN	Contin tion of lication US
19 33027			
			Divisi of app ion US
19869 2330			
			Divi. f app ion US
1990C			
			Conti ion of a plication US
19 6 489			
			Cont of a plication US
1995472295			
			ontin plication US
19 925			
			Divisi of pa US 4967326
			Divisi US 5243698
			onti US 5491359
			Conti of atent US 6414368

Alo Abstract EP A

Th microcomputer comprises a processo , the processor being to execute a number of operat to program instr ns . Each instruction size and has the same format of bit positions, pred function bits designating a required functi , ned positions cont ta designating bits.

Th essor includes a number of egi ected by data transf r appts. one being an operand regis memory has a locat to respectively receive th bits. The transfer appts. includes circuitry struction into the temp memory and for loading b ts temporary

stor...
location into the op... register. ... de... provided for
contr...
the tr... appts. and registers, a... is ... ons... the... function
bits
in th... first temporary memory location.

Equivalent Alerting Abstract US A
The microcomputer system comprises a sin... integrated circuit device
providing a processor and memory.
Th... processor is arranged to execute a ... on data
in
respo... to a program consisting of sev... instructions for
sequ...
execution by the proce...
Ea... instruction ... a set ... which designate
a
requir... function to be e... cuted by the ...essor.
The processor includes several registers ... data transferred to and
from ... registers.
Each instruction is received and load... into one of the
processor registers.
Data transfer and registers responsive ... function bits are
contr... to cause the processor ... oper... with the
funct... bits.
The memory comprises an array of mem... cells ...iding at least
one
K by... RAM for stori... program to ... executed by the processor.
AD... - Improved ... of ... computer.
A... - (43pp)

Equiv... Alerting Abstract US A
A ... is established within the ... the process,
comp... addressable memory locations. A first location
poin...
is co... in a first register employing ... or more second registers
for
stor... variables or other ... ess.
In...tions form the memory are ... in accordance with the
progr..., each instruction having ... data ... a function
port... The function portion has its ... a selected
fun... table ... The ... set
incl...
a t... function ... of the instru... of the data
por...
have... length ... the number of ... which would be required
to
addr... each ... addressable ...
... - Increased ... ed. ... effi... use of
avai...
mem...
...

Equ... Alerting ... US A
Th... computer ...
Pr...

can be stored in the on-chip RAM. One local variable of each process to be
executed is a workspace pointer (WPTR), and each process has a respective
work... identified by the WPTR. For each process, addressing of other
vari...s is relative to the current WPTR ... in a
work...
point... register (WPTR REG). Memory add... are formed by
comb... the contents of ... the ... register and the
oper...
regi...
or ... contents of the ... register and the operand register.
Scheduling and descheduling of processes ... achieved by forming a linked
list within the several workspaces for the ... ive
processes. Each workspace identifies the workspace pointer of the next
process ... be executed. Each workspace contains in memory the
identification of the next instruction ... for that
resp...
pro... Each microcomputer chip can be ... to other such
chips ... respective ... only ... unidirectional
chan... Each channel ... two ... the
iden... and one for data. Communications are synchronised.
(4...

Equivalent Alerting Abstract US A

The microcomputer comprises a memory and a processor to execute a number of concurrent ... and ... are ... them. The
micro... includes ... indi... process as well
as a collection of processes awaiting execution. Processes may be added to
the ... A next process indicator ... indicate the next
process ... be executed.

A ... between
con... or ...
micro... sched... it to the
collection or terminating execution of the current process.
(41...

Equivalent Alerting Abstract US A

The method involves communicating data between process in an array of
comp...

The ... chip ... on-chip RAM
comb...
chan...
and ...

...

syn... an
array

Eac... ice.

Co... ions ... data portions.

Sc... occur by
formi...

active process.
Each workspace identifies the next process to be executed and the next instruction [illegible] process.
ADVANTAGE - Enables computer to operate concurrently with other computers.
ADVANTAGE - (10pp)

Equivalent [illegible]
In the system [illegible] configurations [illegible] microcomputers on an integrated circuit chip with an on-chip memory [illegible] that holds a sequence of instructions for execution by an on-chip processor.
The RAM is protected from noise from on-chip transistors that operate independently of the CPU.
Generally, the configurations involve [illegible] to a substrate, with first and second isolation [illegible], or first and [illegible] isolation walls, [illegible]
ADVANTAGE - [illegible] network of microcomputers with rapid communication between concurrent processes
ADVANTAGE - [illegible]

Equivalent Alerting Abstract US A
The microcomputer includes an integrated circuit device with processor and memory and communication links arranged to provide non-shared connections to similar links of other microcomputers. The links include message synchronisation [illegible] permit creation [illegible] networks of microcomputers.
A circuit controls the data transfer devices [illegible] registers, and responds to function bits [illegible] to [illegible] in accordance with the bits. The function [illegible] or [illegible] functions which cause the control circuit to [illegible] values into the temporary memory of the communications links. This is so that message transmission through the links can be synchronised.
ADVANTAGE - Provides rapid communication [illegible] concurrent processes.

Equivalent [illegible]
A microcomputer comprising an on-chip processor and an on-chip memory on a single integrated [illegible] having [illegible] of semiconductor material of a [illegible] memory comprises storage locations including a high density RAM of [illegible] having at least 1K bytes of storage, said computer including:
(a) an instruction [illegible] circuit for [illegible] instruction storage locations [illegible] instructions [illegible]
(b) an instruction [illegible] circuit [illegible] said instruction storage [illegible] instructions from said locations,
(c) an [illegible] circuit [illegible] said instruction

receiving circuit [illegible] instruction [illegible] said instruction receiving circuit;

(d) a plurality of on-chip transistors [illegible] circuitry operable independently of the operations of said [illegible]

(e) a first isolation region in said substrate, said first isolation region being of the same type of material as that of said substrate and containing all of said memory cells of said high density RAM array;

(f) a second isolation region separate from said first isolation region and being of the same type of material as [illegible] said substrate, said second isolation region containing some of [illegible] transistors which are operable independently of said operation [illegible] and

(g) isolation means [illegible] substrate [illegible] isolating said first and second regions, [illegible] density [illegible] located on the same chip as said independently operable transistors [illegible] is protected from noise due to independent operation of said transistors.

Title Terms/Index Terms/Additional Words: [illegible]; MULTIPLE; REGISTER ; PROCESSOR; ADDRESS; MEMORY ; LOCATE; INSTRUCTION ; BIT; SIZE; FORMAT; REDUCE; [illegible]; DELAY

Class Codes
International Classification [illegible] Attribute
IPC + Level Value [illegible] Revision
G06F-0015/78 A [illegible]
G06F-0015/[illegible] A [illegible]
G06F-0009/30 A [illegible]
G06F-000[illegible]/318 A [illegible]
G06F-0009/34 A I R [illegible]
G06F-000[illegible] A [illegible]
G06F-000[illegible]/16 A [illegible]
G06F-0009/48 A [illegible]
G06F-000[illegible]2 A [illegible]
[illegible]-002[illegible] A [illegible]
H01L-0021/[illegible] A [illegible] R 20[illegible]101
[illegible]-0021/[illegible] A [illegible]
[illegible]-002[illegible] [illegible] R [illegible]
[illegible]F-0015/76 [illegible]
G06F-0009/30 C [illegible]
G06F-0009/318 C [illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
US [illegible] 395[illegible] [illegible] 364DIG001, [illegible]1200 [illegible] 537[illegible] [illegible]0000, [illegible]15000[illegible] [illegible] 123[illegible] [illegible]239000, [illegible]239400[illegible] [illegible]1240[illegible] [illegible] 1241[illegible] [illegible] 364244000, [illegible]244600[illegible] [illegible] 425[illegible] [illegible]255500, [illegible]557[illegible] [illegible] 1267 [illegible] [illegible]271000,

364271300, 364271500, 364280010, 364281, 364281800, 364284000,
364234400, 365000000, 364010001, 364229, 364229100, 364230000,
364230300, 364231400, 364231600, 342328, 364232910, 364232400,
364262510, 364271, 342712, 364281400,
3642, 395325, 35375, 364212800,
395, 36409, 54241, 315300000,
3, 373, 257371, 373 3000,
12014000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F03

...has addressable memory locations and uses instructions having same
bit size and format, with reduced decoding delays

Original Titles:
... function set for a microcomputer...

...microcomputer ... function

Alerting Abstract ...The microcomputer has a processor and a memory, ... to execute ... operations in response to program instructions. Each instruction ... of the same
bit size and has the same format of bit positions, predetermined locations being function ... required function, other
predetermined positions contain data designating bits...

...number of registers ... by data transfer apparatus, one being an
operand register ... has ... to respectively receive the function ... includes circuitry for ... on ... temporary memory and
for ... into the operand register ... transferring the transfer apparatus
and registers, ... the function bits in the first temporary memory ...

Equivalent Alerting Abstract ...microcomputer system comprises a single integrated circuit device providing processor and memory .
...

...processor is arranged to ... number ... operations ... data in response ... or
sequential
execution ...
... of ... designate a
required function ... executed by the processor...

...instruction is ... and loaded into one of the
processor registers...

...Data transfer and reg[illegible] [illegible]ponsive [illegible] function [illegible] are controlled to cause the [illegible] to operat[illegible] in [illegible]ordance with the function bits...

...The memory [illegible] is [illegible] providing at least one K bytes of RAM [illegible] storing [illegible] program to [illegible] executed by the processor...

...A workspace is [illegible] within the [illegible] for the process, comprising several [illegible] memory [illegible]. A first location pointer is contained in a first register employing [illegible] more...

... Instructions [illegible] are [illegible] in accordance with the program, each instruction having a data portion and a function portion. The function portion has bits which indicate a selected function from a set of [illegible] functions [illegible] function set includes a types of function [illegible] the instructions. Each of the data portions has a bit length less than the number of bits which would be required to address directly each of the addressable [illegible] locations...

...ADVANTAGE - Increased operating speed. [illegible] efficient use of available memory space...

...The microcomputer includes [illegible] of RAM, R[illegible], registers and an ALU. Prog[illegible] RAM. [illegible] of each process [illegible]...

...relative to the curr[illegible] WPTR, which is stored in a workspace pointer register (WPTR REG). Memory address locations are formed by combining the contents of the workspace pointer register and the...

...identifies the [illegible] pointer of the [illegible] process to be executed. Each [illegible] memory [illegible] the [illegible] of the next instruction to be executed for the respective process. Each micr[illegible] chip [illegible]...

...[illegible] memory [illegible] execute a number of [illegible] share [illegible]...
...The single chip microcomputer has pr[illegible] stored [illegible] on-chip RAM [illegible] link [illegible] an input channel and an output channel...

...[illegible] bit size instructions have [illegible] and data portions...

...[illegible] the next [illegible] executed and the next

instr... f... its ... s ...

...th system different configurations of ... microcomputers ... an integrated
circuit chip with an on-chip memory use high density RAM that holds a sequence of instructions for execution by an on-chip processor...

...The microcomputer includes an integrated circuit device with processor
and memory and communication links arranged to provide non-shared connection to similar links of other microcomputers...

...A circuit controls data transfer devices and ... and responds to
function bits to ... in accordance with the
bits. The function ... which cause the
control circuit to load pointer values into the temporary memory of the
communications links. This is so that message transmission through the links can be...

...A ... computer ... an on-chip processor and an on-chip memory
on a single integrated circuit chip having a substrate of semiconductor material of a first type ... comprises storage
locations ... at least ...
bytes ... said microcomputer including...

...a an instruction pointer circuit for ... instruction storage
locations ... program instructions therefrom...

...b) an instruction receiving circuit ... said instruction storage locations ... said ... from said
locations...

...c) an instruction ... circuit ... said instruction receiving ... instruction ... said instruction
receiving circuit...

...the same type of material as that of said substrate and containing all
of said ... said high density ...

Title Terms... ... Additional Words ... MEMORY ; ...
... ...

Original Publication ... by Authority

Original Abstracts:
Mic... A computer comprises ... integrated circuit device

having addressable memory locations and a processor with a plurality of
registers (60, 61, 65- [illegible], 69-74) arranged for sequential execution of a plurality of instructions each instruction [illegible] of the same bit size [illegible] format of bit [illegible], one [illegible] of bit positions [illegible] function bits and other bit positions being data bits.
Such instructions [illegible] processor to operate with a reduced number
of registers (6[illegible], 63, [illegible]-67, [illegible]-74) and reduced [illegible] in decoding instructions.

...

...A microcomputer comprises an integrated circuit device with processor
and memory and communication links arranged to provide non-shared connections to similar [illegible] other microcomputers...
...A [illegible] single chip microcomputer includes 4K of
RAM, [illegible] registers [illegible] ALU. Program [illegible] in the on-chip RAM.
The [illegible] local variables [illegible] each process...

...[illegible] to the current [illegible] which is [illegible] pointer
register [illegible] PTR [illegible] G). [illegible] instructions are constant [illegible], having a
function portion and a data portion loaded, respectively, into an
instruction buffer (IB) and an operand register (OREG). Memory
address locations are formed by combining the [illegible] of the workspace
pointer register and the ...

...or [illegible] A pointer and the operand register. A set of
"direct [illegible] from REG [illegible] functions" use the
OREG [illegible] functions [illegible] data from registers
other than the operand [illegible] "prefix [illegible]" instruction (PFIX)
devel[illegible]
oper[illegible] Scheduling [illegible] of
proc[illegible]
are achieved by...

...identifies the workspace pointer of the next process to be executed.
Each [illegible] contains in memory the identification of the next
instruction [illegible] for that respective [illegible] "last
pointer"
regi[illegible] in...
...A [illegible] single chip [illegible] includes 4K of
RAM, [illegible] registers [illegible] Program [illegible] in the on-chip RAM.
The [illegible] local variables [illegible]...

...r [illegible] pointer
regi[illegible] EG,. Instructions are [illegible] having a
func[illegible] loaded [illegible] into an
ins[illegible] operand [illegible] Memory

addr[illegible] by combining [illegible] of the workspace pointer register and the...

...or the contents of the [illegible] register and the operand register. A set of "direct functions " obtain data from OREG. "Indirect functions " use the
OREG [illegible] to identify [illegible] functions [illegible] data from
registers
other than the operand register. A " prefixing " function (PFIX)
develops
operands having long bit lengths. Scheduling [illegible] handling of
processes
are [illegible]...

...identifies the workspace pointer of the next process to be executed.
Each workspace [illegible] in memory the identification of the next
instruction [illegible] for that respective pointers. A "last
pointer"
register (LPTR REG.) [illegible]...
...A programmable, high speed, single chip microcomputer includes 4K
of
RAM, [illegible] registers and [illegible]. Program [illegible] on-chip
RAM.
The [illegible] local variable of each process...

...[illegible] to the current [illegible], which is stored in a workspace pointer
register [illegible]. Instructions are [illegible] a
function [illegible] function loaded [illegible] an
instruction [illegible] an operand [illegible]. Memory
address [illegible] by combining [illegible] of the workspace
pointer register [illegible]

...or the contents of the [illegible] register and the operand register. A set of
"dir[illegible] use
the
OREG contents to identify other functions obtaining data from
registers
other [illegible] operand register. A " prefixing " function (PFIX)
deve[illegible]
oper[illegible] bit lengths. Scheduling and [illegible] handling of
proc[illegible]
are [illegible]...

...identifies [illegible] pointer of the next [illegible] executed.
Each [illegible] the identification of the next
instruction [illegible] for that respective [illegible]
poin[illegible]
reg[illegible]...
...A [illegible] memory and [illegible] is arranged to
exec[illegible]
a plurality of concurrent processes and [illegible] time...

...S[illegible] program [illegible] on-chip RAM
con[illegible] links [illegible]
cha[illegible]
an[illegible]

...m[illegible] computer [illegible] Each process[illegible] has a workspace. Const[illegible] bit[illegible] have function [illegible] data portions. Sched[illegible] in each microcomputer occur by form[illegible]
a li[illegible]...

...A microcomputer comprises an integrated circuit device with pro[illegible]
and [illegible] and communication links arranged to provide non-shared connections to similar links of other microcomputers...

...A [illegible] comprises an integrated [illegible] device with proc[illegible]
and memory and communication links arranged to provide non-shared conn[illegible] other micro[illegible]...

...A [illegible] integrated [illegible] with proc[illegible]
and [illegible] communication links arranged to provide non-shared conne[illegible] to similar links of other microcomputer...

...A microcomputer comprises an integrated circuit device with proc[illegible]
and memory and communication links arranged to provide non-shared connections to similar links of other microcomputer...

...A [illegible] integrated [illegible] with proc[illegible]
and [illegible] links [illegible] non-shared conn[illegible] other micro[illegible]
Clai[illegible]
The microcomputer comprises a processor and a memory, the processor bein[illegible]
able to execute a number of operations in response to program instr[illegible] instruction is of the same [illegible] size and has the
sa[illegible] bit positions, predetermined [illegible] being function bits indicating a required function, other predetermined positions cont[illegible]...

...n[illegible] of registers interconnected by data transfer appts. one being an
oper[illegible] memory [illegible] respectively rec[illegible] The [illegible] includes cir[illegible] for [illegible] instruction [illegible] memory and
for [illegible] bits [illegible] temporary [illegible] into the oper[illegible] transfer ap[illegible]
an[illegible] to the [illegible] first te[illegible] memory[illegible]...

...[illegible] operating a process in a mi[illegible] having proc[illegible]
m[illegible] addressable memory locations [illegible] program defining instr[illegible] said method [illegible] listing wit[illegible]

the ... comprising a
plur... memory lo...
lo...
the ... establishing a
fir...

...a ... workspace, ... first register
co... said ... pointer, ... said
m...
ch... the same bit
leng... speed of
operation and improve ... of memory, ... of said
instructions
having a data portion and a function portion selected from a set of
selectable functions, (b) each of said data portions having a bit
leng...
less ... the number of bits which would be required to address
dir...
each ... memory location, (c) in response to an
instr... type, combining ... said first
reg...
with ... data ... said instruction ... memory
ad...
a...
tran...
info... memory address, ... in response to an
instr... of a second type, combining ... of ... second
regis...
with ... data portion of said instruction ... second type to
defi...
an a... memory location and perf... transfer of
inf...
usi... said first ... registers and each
of
said ... locations having ... of
ho...
th... addressable ...

...<... device
havi...
a pr... integrated
circ...
dev..., and a plurality of...

...comprising a ... integrated circuit
de...
wh... having a plurality
o...
instr... circuit
d...

... a ...
co...
pr... instr...
in...g
a ... channel
o...

the f... or second type depending on whether...
...the ... integrated circuit device or ... distinct integrated circuit
device, the sequence of instructions in ... program incorporating said
co... instructions being the same ... type of
char... addr...

...A microcomputer ... on-chip ... an on-chip memo
on a single integrated circuit chip having ... substrate of semiconductor material of a first type, wherein said on-chip memory comprises stor...
locations including a high density RAM of memory cells having at least ...
bytes ... said ... instruction
pointer circuit for addressing instruction storage locations to obta...
program instruction therefrom; (b) an instruction receiving circuit
coupl... said instruction storage locations ... receiving said
instructions from said locations; (c) an instruction decoder circuit
coupled to said instruction receiving ... decoding instructions
recei... said instruction receiving circuit; (d) a plurality of
on-chip ... circuitry ... independently of the...

...the ... type of ... containing all
of ... memory cells of said high density RAM ... (e) a second
is... from said...
... circuitry and an
on-chip ... memory ... on a single ... chip having a
sub... of semiconductor material ... wherein said on-chip
write ... memory comprises a high density array of memory cells having
at ... bytes of ... said ... chip including
(a) ...
memory ... circuitry ... said ... plurality of
additional ...
on-chip ... circuitry ... independently of the
op... said
ch... memory cells
o...
said high density ... array; (d) a ... on said
chip ...

... some of said additional ... operable
independently ... operation of said ... (e) an
iso... said first ...,
said
s... located an...
in...
wh... array ... chip as

said [illegible] independently [illegible] additional transistors and [illegible]...

...An integrated circuit chip comprising an on-chip processor, an on-chip ROM holding micro instructions for the processor and an on-chip writable memory on a single integrated-circuit chip having a substrate of semiconductor material of a first type, wherein said on-chip writable memory comprises a high density array of memory cells having at least one K [illegible] of storage, said integrated circuit chip including (a) memory array [illegible] on said chip; (b) a plurality of additional on-chip transistors [illegible] operable independently of the operation of said memory array; (c) a first isolation region on said chip, said first isolation region containing all of said memory cells of said high density memory array; (d) a second isolation region on said chip separate from said first isolation region...

...containing some of said additional transistors which are operable independently of the operation of said memory array, and (e) an isolation interface [illegible] said first and second isolation regions, said second [illegible] isolated and [illegible] isolation interface whereby said high density memory array is located on the same chip as said independently operating additional transistors and [illegible]...

...An integrated circuit chip comprising on-chip [illegible] memory and an on-chip writable memory on a single integrated circuit chip having a substrate of semiconductor material wherein said on-chip writable memory comprises a high density array of memory cells having at least 1 K bytes of storage, said integrated circuit chip including [illegible] memory array and [illegible] said memory [illegible] operable to [illegible] on-chip writable memory [illegible] at least one [illegible] for connection to off-chip...

...comprising a plurality of additional on-chip transistors operable independently of the operation of said memory array [illegible] first isolation region [illegible] a first isolation region [illegible] memory [illegible] memory array; (d) a second isolation region on said chip separate from said first isolation region, containing some of said [illegible] are operable [illegible] said [illegible] memory array and (e) an isolation interface region between said first and second isolation regions, whereby said high density...

memory [illegible] is located on the same chip as said independently
operat[illegible];
additional [illegible] and is...

```
Set     Items   Description
S1    1471788   C[illegible]AND? ? OR INSTRUCTION? ? OR PROGRAM? OR PROGRAMME?
? OR
                 CODE? OR CODING? OR FUNCTION?
S2     378164   S1(5N)(MERG??? OR FUSE? ? OR FUSING OR UNIFY? OR
UNIFI? ?
               OR UNITE? OR SYNTHESI? OR COMBIN? OR INTEGRAT? OR INCLU?
OR I-
               NCORPORAT?)
S3    1227831   ENTRY? OR PREDICAT? OR PREFIX? OR PRE()(FIX???) OR
SUFFI[illegible]
               OR M[illegible] OR MODIF?
S4     275954   S2 AND S3
S5      3[illegible]061   S4(5N)(STORE? ? OR STORING OR SAVE? ? OR ACCUMULAT? OR
KEE-
               P??? OR WRIT??? OR UPDAT?)
S6     [illegible]   BUFFER? OR MEMOR? OR CACHE? OR CACHING? OR QUEUE?
S7     102649   S[illegible](3N)(SPECIF? OR INDICAT? OR DESIR? OR REQUIR? OR
NECESS?
               OR CERTAIN?)
S8     22[illegible]63   (WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR
BITS))(5N)(EQUAL? -
               OR AT()LEAST OR COMPARABL? OR IDENTICAL? OR EQUIVALEN? OR
SAME
                OR SIMILAR)
S9     2[illegible]   (WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR
BITS) (5N MATCH?
               OR [illegible] OR ALIKE OR AKIN OR CONGRUEN? OR COMMON? OR
INCOMMO-
               N?
S10     [illegible]   [illegible](5N)(WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR BITS))
S11     [illegible]   S[illegible](5N)(WIDTH? OR SIZE? ? OR NUMBER(2N)(BIT OR BITS))
S12         33   S5(100N)[illegible]7(100N)S8:S9(100N)S10:(100N)[illegible]
S13         20   [illegible] 2007 OR AY=[illegible]:2007)
File 348:EUROPEAN PATENTS 1978-2007/ 200[illegible]
        (c) 2007 European Patent Office
File 349:PCT FULLTEXT 1979-2007/UB=20070913UT=2007[illegible]
        (c) 2007 WIPO/Thomson
```

13/5,K/? (Item ? from file: 348)
DIALOG(R) File 348: EUROPEAN PATENTS

02055?69
Routing and forwarding table management for network processor architectures
Verwaltung von Routing- und Weiterleitungstabellen fur Netzwerkprozessorarchitekturen
Gestion d'une table d'acheminement et de reacheminement pour des architectures de processeur de reseau
PATENT ASSIGNEE:
Ra[illegible] Communications Software Division, [illegible] (4440140), 1500 NW [illegible] Street, Des Moines, IA 50325, (US), (Applicant designated States: all)
INVENTOR:
Sch[illegible], Curt, 1500 NW 118th Street, Des MoinesIA 50325, (US)
Lyons, David, 1500 NW 118th Street, Des MoinesIA 50325, (US)
Stern, Rick, 1500 NW 118th Street, Des MoinesIA 50325, (US)
LEGAL REPRESENTATIVE:
Patentanwalte [illegible]stetter, Schurack & Skora (102941), Balanstrasse 57, 81541 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 1657633 A1 06 517 [illegible]
APPLICATION (CC, No, Date): EP 200[illegible] 021[illegible];
PRIORITY (CC, No, Date): [illegible] 011[illegible]
DESIGNATED STATES: AT; [illegible]; CH; CY; CZ; DE; DK; [illegible]; FI; FR; GB; GR; HU; IE; IS; [illegible]; LT; [illegible]; MC; NL; PL; PT; RO; SE; SI; SK; TR
EXTENDED DESIGNATED STATES: AL; BA; HR; MK; YU
RELATED PARENT NUMBER(S) - PN (AN):
EP [illegible] (EP [illegible])
INTERNATIONAL PATENT CLASS (V8 + ATTRIBUTES):
IPC + Level Value Position Status Version Action Source Office:
G0[illegible] [illegible] B 20060101 20060322 H EP

ABSTRACT EP 1657[illegible]
A [illegible] system having a network processor comprising a core processor and at least one microengine in operative communication with the [illegible] processor is provided. A table comprised of a plurality of entries with [illegible] associated therewith is built, wherein the entries are [illegible] hierarchically according to an LC-Trie [illegible] addresses. An information packet is [illegible] with [illegible] system, wherein the information packet has [illegible] associated therewith. The table is searched using an LC-Trie [illegible] algorithm to find a match between an IP address of an entry [illegible] destination IP address [illegible] information packet [illegible] is transmitted [illegible] IP address [illegible] of the matching entry [illegible] interface is

provided to accommodate communication between the core processor and mi... engine of the network processor.
ABSTRACT WORD COUNT: 145
NOTES:
Figure number on first page: 4

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: ...151 A1 Published application with search report
Change: ...614 A1 Title of invention (German) changed: 20060614
Change: ...614 A1 Title of invention (English) changed: 20060614
Change: 060614 A1 Title of invention (French) changed: 20060614
Change: ...613 A1 Title of invention (German) changed: 20060613
Change: 060613 A1 Title of invention (English) changed: 2006...
Change: ...613 A1 Title of invention (French) changed: 2006...
Change: ...108 A1 Title of invention (German) changed: 2006...
Change: ...108 A1 Title of invention (English) changed: 2006...
Change: ...108 A1 Title of invention (French) changed: 2006...8
Change: 070124 A1 Title of invention (German) changed: 2007...
Change: 070124 A1 Title of invention (English) changed: 2007...
Change: 070124 A1 Title of invention (French) changed: 2007...
Change: ...125 A1 Title of invention (German) changed: 2007...
Change: ...125 A1 Title of invention (English) changed: 2007...
Change: ... A1 Title of invention (French) changed: 2007...
LANGUAGE (Publication, Procedural, Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200620	1484
SPEC A	(English)	200620	14137
Total word count - document A			15621
Total word count - document B			0
Total word count - documents A + B			15621

...SPECIFICATION ... terminal leaf node... shown in Table 3,
... would be ... entry of the ... the number of ... being ... Build function;
... the number of bits in any common prefix among the set of ... being processed by the Build function; and ... - the first...

...Skip function that examines the first and last entry in the

sub[illegible]val for a [illegible]mmon prefix of bits, the number of bits in any such prefix comprises the Skip value. In the initial call to the Build function there is no...

...a [illegible]uteBr[illegible] function that examines [illegible] of [illegible] in the subinterval to determine the [illegible]gest number of prefix [illegible], disregarding any com[illegible] prefixes that contain all of the [illegible] values [illegible] that numb[illegible] of [illegible]. Using the first...

13/5,K/2 [illegible] from file: 348
DIALOG(R)File [illegible]

01930027
Secure transaction management
Verfahren und Vorrichtung zur gesicherten Transaktionsverwaltung
Procede et dispositif de gestion de transactions securisees
PATENT ASSIGNEE:
  Intertrust Technologies Corp., (2434323), 955 Stewart Drive, Sunnyvale,
    CA 94 85, (US), (Applicant designated States: all)
INVENTOR:
  Ginter, Karl [illegible], 10404 43rd Avenue, Beltsville, [illegible], (US)
  Spahn, Francis [illegible], [illegible] Avenue, El Cerrito [illegible] 9453[illegible], (US)
  Shear, Victor [illegible], [illegible] Lane, Bethesda, [illegible], (US)
  Van Wie, David [illegible], [illegible] Williametre [illegible], 6, Eugene, OR 97401, (US)
LEGAL REPRESENTATIVE:
  [illegible], [illegible] Davis Lewis (28273), BERESFORD & Co. [illegible] 13 High Holborn,
    London WC1V 6BX, (GB)
PATENT (CC, No, Kind, Date): EP 1555591 [illegible] 05[illegible] (Basic)
                             EP 1555591 [illegible] 051[illegible]
APPLICATION (CC, No, Date): EP 2005075672 960213;
PRIORITY (CC, No, Date): US 388107 [illegible] 213
DESIGNATED STATES: [illegible]; CH; DE; [illegible]; ES; [illegible]; LI; LU; MC;
  NL; PT; SE
RELATED PARENT [illegible] (AN):
  EP 861461 (EP [illegible]371)
INTERNATIONAL PATENT CLASS (V7): G06F-001/0[illegible]; G06F-0[illegible]/60

ABSTRACT EP [illegible]
  [illegible] method of and apparatus for assembling software elements to form a
  component assembly [illegible] are described. [illegible] record [illegible] containing
  information identifying the software elements [illegible], 1200, 1202,
  69[illegible], to be assembled to form the component assembly is accessed. At least
  [illegible] of the [illegible] elements (10[illegible], 1100) [illegible] by the record
  [illegible] executable program code and at least [illegible] the software
  elements is a [illegible] (1100) compris[illegible] executable program code and a
  header [illegible]04) having an execution space identifier identifying which of a
  [illegible] levels [illegible]
  assembly [illegible]
  [illegible]. The software elements identified by the record are
  assembled [illegible] assembly [illegible], be loaded
  and executed [illegible] the level of security of the component assembly
  [illegible] level [illegible] the
  ex[illegible] identifier.
ABSTRACT WORD COUNT: [illegible]

NOTE:
Figure number on first page: 23

LEGAL STATUS (... Pub. Date, Kind, Text):
Application: ... A1 Published application without search report
Search report: ... A2 Separate publication of the search report
Change: ... Inventor information changed: ...1018
Change: ... A2 Title of invention ... changed:
2006...
Change: ... A2 Title of invention (English) changed:
2006...
Change: ...0214 A2 Title of invention (French) changed:
2006...
Change: 060726 A2 Title of invention (German) changed:
2006...
Change: ... A2 Title of invention (English) changed:
2006...
Change: ... A2 Title of invention (French) changed:
2006...
Change: ... A2 Title of invention (German) changed:
2007...
Change: ... A2 Title of invention ... changed:
2007...
Change: ... A2 Title of invention ... changed:
2007...
Change: ...070725 A2 Title of invention (German) changed:
2007...
Change: 070725 A2 Title of invention (English) changed:
2007...
Change: ... A2 Title of invention (French) changed:
200...
LANGUAGE (Publication, Procedural, Application): English; English;
English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A ... ...29 ...
SPEC A ... ...529 ...
Total word count - document A ...
Total word count - document B 0
Total word count - documents A + B 195030

... NOTIFICATION and rescore pkt (BYTE *buff ..., ...far
... ...void)
This function ... a packet stored in buffer of ... size.
It
returns 0 if the packet ... successfully, ...

... functions ... for
example, ... manager ...
... in this ... scenario ... service
74...
... a provisioning ... is provided to
... VDE ... A pre-existing ... integrated
with
... an enhanced set of ... calls
... functions ... functions
60...

other than VDE functions (see Figure 11A). The add-on portion of API
service 742 may translate VDE function...

13/5,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01613119
Method and apparatus for monitoring the performance of a computer system
Verfahren und Vorrichtung zur Leistungsuberwachung eines Rechnersystems
Methode et dispositif de surveillance de la performance d'un systeme d'ordinateur
PATENT ASSIGNEE:
Sun Microsystems, Inc., (2616592), 4150 Network Circle, Santa Clara, California 95054 (US), (Applicant designated States: all)
INVENTOR:
Cantrill, Bryan, [illegible] 20th Street, San Francisco California 94107, (US)
LEGAL REPRESENTATIVE:
Davies, Simon Robert et al (75453), D Young & Co, 21 New Fetter Lane, London, EC4A 1DA, (GB)
PATENT (CC, No, Kind, Date): EP 1331566 A2 030730 (Basic)
APPLICATION (CC, No, Date): EP 2003250416 030[illegible]
PRIORITY (CC, No, Date): US 60[illegible] 02[illegible]129
DESIGNATED STATES: AT; BE; BG; CH; CY; CZ; DE; DK; EE; ES; FI; FR; GB; GR;
HU; IE; IT; LI; LU; MC; NL; PT; SE; SI; SK; TR
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; [illegible]
INTERNATIONAL PATENT CLASS (V7): G06F-011/34
ABSTRACT EP 1331566 A2
A system and method are provided for monitoring the performance of a computer system by dynamically interposing an instrumented trap table. A base address of a trap table, which may be contained in a trap base address register, [illegible] be changed to indicate an instrumented trap table.
An instrumented trap table may gather a variety of statistics, including the type of trap and an entry timestamp. An instrumented trap table may then call a non-instrumented trap table to process the trap. A non-instrumented trap table may pass control back to the instrumented trap table to [illegible] statistics, for example an exit timestamp.
An instrumented trap table may then return process flow to the calling routine. [illegible] useful performance statistics may be gathered [illegible] regular [illegible]ware.
ABSTRACT WORD COUNT: [illegible]
NOTE:
Figure number on first page: 1

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: [illegible] Published application [illegible] search report
LANGUAGE (Publication, Procedural, Application): English; English; English;
En[illegible]

FULLTEXT AVAILABILITY:

Available Item		Update	Word Count
CLAIMS A		...0331	1352
...A		...0331	4811
Total word count - document A			6163
Total word count - document B			0
Total word count - documents A + B			6163

...entry, and the ... handler will be accumulated in a memory location specific to ... misses. In order to minimize the amount of knowledge the kernel must have...

...of the TLB return entry. Rather, it executes a known instruction ... a given memory ... with the %tpc in a ... register, for example register ... present...

...core)INSTR ("add ..."). Prior to execution, the ... mechanism may modify this instruction such that ... size of the TLB ... misses from kern... user employs a...

...central processor 605 functionally coupled with the bus for processing information and instructions, a volatile memory 615 (e.g., random access memory ...) coupled with the bus ... information and instructions for the central processor 605 and a non-volatile memory 610 (e.g., read only memory ROM) coupled with the bus ... for storing static information and instructions for the processor ... Computer system ... also ... includes a changeable non-volatile memory ... storing information and ... for the central processor ... which...

...storage device ... (e.g., a rotating magnetic disk) coupled with the bus ... storing information and instructions ... optional ... 630. Device...

13/. [illegible] (Item 4 [illegible] file: 348)
DIA[illegible] File [illegible] EUR[illegible] PATENTS

0151[illegible]
VIDE[illegible] METHOD [illegible]
VIDE[illegible]KUNG[illegible] -VORRICHTUNG
PROC[illegible] DE CODAGE VIDEO ET DISPOSITIF DE CODAGE [illegible]
PAT[illegible] ASSIGNEE:
Koninklijke Philips Electronics N.V., (200[illegible]), G[illegible]wedseweg 1, 56[illegible]
[illegible] Proprietor designa[illegible] all)
INVENT[illegible]:
RU[illegible], Prof. H[illegible]laan [illegible], NL-5656 [illegible] Eindhoven, (NL)
LEG[illegible]:
[illegible], Christian [illegible] (4438[illegible]), Philips Int[illegible]lectual Property & [illegible]andards, 156 Boulevard Haussmann, 750[illegible] Paris, (FR)
PATENT [illegible], No, Kind[illegible]: EP 137[illegible]95 A[illegible]
EP 137[illegible]95 B[illegible]
WO 2002080565 02[illegible]
APPL[illegible]: EP 200[illegible]708586 [illegible] 200213997
02[illegible]
PR[illegible] FR 2001400816 01032[illegible]; [illegible] 011130
DE[illegible] CH; CY; DE; DK; ES; F[illegible]; GB; G[illegible]; IE; IT;
LI;
[illegible]
EXTE[illegible]SIGNATED [illegible] AL; LT; [illegible]; MK; [illegible]
IN[illegible] 007/[illegible]
IN[illegible] (V8 + ATTRIBUT[illegible]):
IPC [illegible] Val[illegible] Status Version A[illegible] Office:
H04N [illegible] 20060101 [illegible]
NOTE:
No A[illegible] EPO
LEGAL [illegible] (Type, Pub Date, Kind, Text):
App[illegible] International applic[illegible] (Art. 158(1))
App[illegible] International applic[illegible] entering European phase
App[illegible] Published application [illegible] search report
Exa[illegible] date of request [illegible] 031029
E[illegible] Date of dispatch [illegible] examination report: [illegible]
Ch[illegible] Title of invention [illegible] changed:
20[illegible]
Ch[illegible] Title of invent[illegible] changed:
20[illegible]
Ch[illegible] Title of invent[illegible]:
200[illegible]
Gra[illegible] B1 Granted patent
Chan[illegible] Title of invent[illegible] changed:
20[illegible]
Change: [illegible]214 B1 Title of invention (E[illegible]) changed:
2[illegible]
C[illegible] Title of invent[illegible] changed:
20[illegible]
C[illegible] Title of invent[illegible] changed:
2[illegible]
C[illegible] Title of inven[illegible] changed:

Change: Title of invention changed:
Change: 070123 Title of invention changed:
Change: Title of invention changed:
Change: 070123 Title of invention (French) changed:
Change: Title of invention changed:
Change: Title of invention changed:
Change: Title of invention changed:
Change: Title of invention changed:
Change: 070111 Title of invention changed:
Change: Title of invention changed:
Change: 070 Title of invention changed:
Change: Title of invention changed:
Change: Title of invention changed:
Change: Title of invention changed:
Change: Title of invention changed:
Change: 070519 Title of invention changed:
LANGUAGE (Publication Procedural,Application): English; English
FULLTEXT AVAILABILITY

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200638	
CLAIMS B	(German)	200638	
CLAIMS B	(French)	200638	600
SPEC B	(English)	200638	

Total word count - document A
Total word count - document B 420
Total word count - documents A + B 420

... SPECIFICATION ... P frames, and I frames ... bitstream is ... in an encoder ... buffer ... either ... buffer of a ... bits ... from ... I, P, B frames can ... the quantizer ... each ... quantization ... its, for a ... quantizer...

...analysis step is followed by a second pass which processes said statistics in order to modify at least the quantizer step size and, thus, to perform a more harmonious distribution of bits for each macroblock of the...

...from ... while insuring that a minimum channel ... needed so as
to ... buffer ... at the decoder side ... bitrate is
obtained, and the MPEG-2 standard...

...coefficient linearly depending on the ... corresponding to the
concerned macroblock. The complexity is equal to the product of the number of bits used for coding each macroblock and the quantization step size.

...according to an ... of...

...leading to an output bitstream ... is ... during which the quantization step size is modified for each macroblock of the frame, in order to modify ... allocation...

...effect detector ..., described hereinunder ... yields a blocking artifact ... which is stored and used in order to modify
the statistics (complexities in the present case), now referenced by ST... in Fig. 2. ...

13/3,K/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

0129[illegible]

INITIAL STAGE OF A MULTI-STAGE ALGORITHMIC PATTERN GENERATOR FOR TESTING IC CHIPS

ANFANGSSTUFE EINES MEHRSTUFIGEN ALGORITHMISCHEN MUSTERGENERATOR ZUR PRUFUNG VON IC-BAUSTEINEN

ETAGE INITIAL D'UN GENERATEUR ALGORITHMIQUE A PLUSIEURS ETAGES DE MOTIFS MULTIPLES POUR TESTER DES MICROCIRCUITS INTEGRES

PATENT ASSIGNEE:
UNISYS CORPORATION, (84[illegible]94), Township Line and Union Meeting Roads P.O. Box 500, Blue Bell, PA 19424-0001, (US), (Proprietor designated states: all)

INVENTOR:
[illegible], James, Vernon, 6462 W. Victoria Lane, Chandler, AZ 85226, (US)
[illegible], Robert, Davis, [illegible] E. Flintlock Place, Chandler, AZ 85249, (US)

LEGAL REPRESENTATIVE:
[illegible] Guido Dr.-Ing. et al [illegible] Tiedtke-Bühling, Kinne & Partner [illegible] Munchen (DE)

PATENT (CC, No, Kind, Date): [illegible] A1 [illegible]
EP [illegible]
EP [illegible]
WO 2001033235 [illegible]

APPLICATION (CC, No, Date): EP 2000975345 [illegible]; WO 2000US29265 [illegible]

PRIORITY (CC, No, Date): US 432969 [illegible] 1103

DESIGNATED STATES (Pub A): AT; BE; CH; CY; [illegible] GB; GR; IE; IT; [illegible] PT; SE; (Pub [illegible]): DE; [illegible]

EXTENDED DESIGNATED STATES: AL; LT; [illegible] MK; [illegible]

INTERNATIONAL PATENT CLASS (V7): [illegible]

CITED PATENTS (EP [illegible]): EP 5[illegible] A; US 43[illegible] A; US [illegible] A; US 528[illegible]

CITED PATENTS [illegible] A; US 43[illegible]; US 5[illegible] A; US 43[illegible]

NOTE:
No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: [illegible] A1 International application (Art. 158(1))
Application: [illegible] 04 A International application entering European phase
Application: [illegible] 31 [illegible] Published application [illegible] search report
Examination: [illegible] 31 [illegible] Date of request for examination: [illegible] 020423
Change: [illegible]
200[illegible]
Change: [illegible]
20[illegible]

Cha[illegible] Title of inve[illegible] [illegible]ench) [illegible]
2004[illegible]
Grant: 040915 B1 Granted patent
Change: 040901 A1 Title of inventi[illegible] (French) changed:
2004[illegible]
Change: 040[illegible]01 A1 Title of inventi[illegible] (English) changed:
2004[illegible]
Change: [illegible]01 A1 Title of inventi[illegible] ([illegible]) chang[illegible]:
200[illegible]
Grant: 040915 B1 Granted patent
Op[illegible]: [illegible] B1 No opposition filed: 20[illegible]
Cha[illegible]: [illegible] B1 Title [illegible] inventi[illegible] ([illegible]) changed:
2006[illegible]
Chan[illegible]: [illegible] B1 Title [illegible] inven[illegible] (English) changed:
2006[illegible]
Ch[illegible]: [illegible]18 B1 Title of inven[illegible] (French) changed:
200[illegible]
Cha[illegible]: [illegible]09 B1 Title of inventi[illegible] (German) changed:
2006 [illegible]
Cha[illegible]: [illegible]09 B1 Title of inventi[illegible] (English) changed:
200[illegible]
Cha[illegible]: [illegible]09 B1 Title of inventi[illegible] ([illegible]) ch[illegible]:
200[illegible]
LAN[illegible] (Publication,P[illegible]dural,Application): English; English;
Eng[illegible]
FU[illegible]
Ava[illegible] [illegible] Text Language Update Word [illegible]

[illegible]S B	(English)	[illegible]	[illegible]
[illegible]MS B	([illegible])	[illegible]0438	[illegible]
[illegible]AIMS B	([illegible])	200438	[illegible]
[illegible]EC B	(English)	200438	[illegible]

Tot[illegible] word count - Document A 0
Total word count - Document B [illegible]249[illegible]
Total word count - Documents A + B [illegible]249[illegible]

...S[illegible] [illegible] [illegible]tion [illegible] [illegible]es a[illegible] [illegible]ration code
C[illegible] which identifies it as the write memory instruction, and it
i[illegible] [illegible] fie[illegible] [illegible] Then [illegible] is executed by
th[illegible] pattern generator 1....

...c[illegible] constr[illegible] of TTL circuits, EC[illegible] circuits, MOS circuits, and CMOS
circuits. Likewise, each memory module [illegible] of Figure 1 can be
constructed of memory cells of any type, [illegible]s those which store data
bits [illegible] flip flops....

...[illegible] in c[illegible] with Figure 10.
[illegible] modification, the number [illegible] bits which are rea[illegible]
[illegible] [illegible]de [illegible]d
n[illegible]
F[illegible] [illegible] in the [illegible] memory module 13 consists of "X"
b[illegible] [illegible] each...

...Each TMS header in TABLE 1A is followed [illegible] a series of "1" bits which

equals the [illegible] bits that ar[illegible] the

I[illegible] the number [illegible] bits [illegible] are writ[illegible]

in [illegible] low [illegible] a...

se[illegible], then instruction 33 is referr[illegible] over instruction 37 bec[illegible]

in [illegible] less [illegible] mem[illegible].

[illegible] another modification, any one of the [illegible] streams TDI, ETDO, and MASK

in...

13/ ,K/3 (Item 6 fr m file: 34

0129

SYSTEM FOR TESTING IC CHIPS SELECTIVELY WITH STORED OR INTERNALLY GENERATED BIT STREAMS

SYSTEM ZUM SELEKTIVEN TESTEN EINES IC CHIPS MIT GESPEICHERTEN ODER INTERN ERZEUGTEN BITSTROMEN

SYSTEME POUR TESTER DES PUCES DE CIRCUIT INTEGREES SELECTIVEMENT AVEC DES TRAINS DE BITS MEMORISES OU GENERES EN MODE INT NE

PATENT ASSIGNEE:
UNISYS CORPORATION, (342794), Township Line and Union Meeting Roads P.O. Box 500, Blue Bell, PA 19424-0001, (US), (Proprietor designated states:)

INVENTOR:
RHODES, James, Vernon, 6462 W. Victoria Lane, Chandler, AZ 85226, (US)
CONKLIN, Robert, David, 2267 E. Flintlock Place, Chandler, AZ 85249, (US)
BARR, Timothy, Allen, 243 S. Criss St., Chandler, 85216, (US)

LEGAL REPRESENTATIVE:
Modiano, Guido, Dr.-Ing. (40786), Modiano, Josif, Pisanty & Staub, ... Munchen, (DE)

PATENT (CC, No, Kind, Date): EP ... A1 ... (Basic)
EP 1... 446 ...
WO 20... 0332...

APPLICATION (CC, No, Date): EP 2000973110 ...1024; WO 2000US29302 0010

PRIORITY (CC, No, Date): US 432966 991103

DESIGNATED STATES (Pub A): AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE; (Pub B): DE; FR; GB

EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI

INTERNATIONAL PATENT CLASS (V7): G01R-031/31

CITED PATENTS (EP B): EP 4 ...876 A; ... 57006 ... 52460 A

CITED ... EP ... 4...; ... 400 ...

NOTE:
No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 01 ... International application. (Art. 158(1))
Application: 010704 A1 International application entering European phase
Application: 020731 A1 Published application with search report
Examination: 020731 A1 Date of request for examination: 20020423
Examination: 0... A1 Date of dispatch of the first examination report: ...
Grant: ... B1 Granted patent
Opposition: ... No opposition filed: ... 025
Opposition: ... No opposition filed: ... 025
Lapse: ...7 ... a ... state ... date): DE

Chan :   060405 B1 Title of invention (German) changed:
2006
Change: 060405 B1 Title of invention (English) changed:
2006
Change: 060405 B1 Title of invention (French) changed:
2006
LANGUAGE (Publication,Procedural,Application): English; English;
English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	[illegible]	833
CLAIMS B	(German)	[illegible]	[illegible]
CLAIMS B	(French)	[illegible]	[illegible]
SPEC B	(English)	200619	20597

Total word count - document A
Total word count - document B   22397
Total word count - documents A + B   22397

...SPECIFICATION single word WC. Instruction 36 includes an operation code
OP which identifies it as the write memory instruction , and it includes a memory address field 36a. When instruction 36 is executed by
the [illegible] ...

...can be constructed of TTL circuits, ECL circuits, MOS circuits, and CMOS
circuits. Likewise, the memory module 13 in Figure 1 can be
constructed of memory cells of any type, such as those which store data
[illegible] flip-flops or...

...from running mode as described in conjunction with Figure 10.
[illegible], as another modification , the [illegible] bits which are read
[illegible] memory module 13 can [illegible] numbers.
Figure [illegible] shows that each word in the memory module 13 consists of "x"
bits [illegible] integer number of [illegible]...

...[illegible] of "1" bits which
[illegible] the number of TDO bits that [illegible] from the
[illegible] REGISTER, or the number of TDI bits that are written
in [illegible]...
[illegible] 37 be [illegible]
[illegible]
[illegible] TDO, and MASK
in...